2009

# A Modular Approach to Lung Nodule Detection from Computed Tomography Images Using Artificial Neural Networks and Content Based Image Representation

Omer Muhammet Soysal
*Louisiana State University and Agricultural and Mechanical College*, omsoysal@lsu.edu

# A MODULAR APPROACH TO LUNG NODULE DETECTION FROM COMPUTED TOMOGRAPHY IMAGES USING ARTIFICIAL NEURAL NETWORKS AND CONTENT BASED IMAGE REPRESENTATION

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

In

The Department of Computer Science

by
Ömer M. Soysal
B.S., Hacettepe University, 1992
M.S., İnönü University, 1998 & Southern University, 2002
May 2009

to my mommy and daddy
and
the good people who dedicated themselves to save life

(Biricik anneciğime ve muhterem babacığıma
ve
ömrünü yaşam kurtarmaya adayan iyi insanlara ithaf ediyorum)

"Spend your wealth for the cause of Allah (The God), and cast not yourselves to ruin with your own hands; and do good. Lo! Allah loves the good doers."
(The Holy Quran, 2: 195)

# ACKNOWLEDGEMENTS

All praise to the Lord of the whole universe.

My mommy and my daddy; how can I appreciate both of you adequately, if there is any means? Days and nights; you guide me and love me, you are always with me, and you take care me and support me. Oh my mommy; I am so sorry to be far away from you, years has just passed through my heart leaving tears behind; I beg your forgiveness. You are my guidance for honesty and hard work; the first teacher who enlighten my mind. My daddy; you are always ahead of us considering our future and if my hands are full of skills, that is because of my first master, daddy. I got love in books within your library.

My wife and my kids, Nazan Hatice, Ahsennur and Hanif Rauf; long years to bear my absence as a husband and a father, I hope I may have another chance to be together hereafter. My brother, Ahmet Fatih; full of life and happiness, I am looking forward days to hug my nieces. My lovely grandmothers, aunts and uncle, Emine, Rukiye, Gadiriye, Sevinç, and Necati; I missed you so much. Mehmed Said Çekmegil, my granduncle; he is my first teacher who taught how to question and how to criticize; he never gave a fish, you must learn to catch it. My professor Dr. Ercan Oztemel; I appreciate his help and encouragement to pursue my PhD in USA. My friends, Ahmet Tozal, Dr. Osman Kandara, Ahmet Tan, Kemal Birtek, Dr. Cevdet Akbay, and Dr. Necati Engec; it is a great blessing being with them far away home.

Dr. Jianhua Chen, my major professor; I cannot forget the time she accepted to be my supervisor in PhD, she saved my PhD dream; she criticizes with challenge to open the mind to horizons but does not discourage, she suggests but does not force, she seeks perfectness but does not ignore reality. Dr. Helmut Schneider, my minor professor; you never know if talking to your boss or your friend, what a generous person, he always motivates and supports, he is amazingly full of productivity; but be careful, he cares quality of work, never be around if you do not care quality. My dissertation committee; how could I be luckier than having such a great committee? Dr. Steven Bujenovic; it would be a nightmare if I did not meet with him to accomplish this work; I hope we will continue to work together. Dr. Bahadir Gunturk; he is so friendly but serious and full of practicality, I thank his valuable discussions, especially, on image representation. Dr. Kenneth L. (Kip) Matthews II; he is calm and but friendly, I appreciate his valuable discussions, especially, on assessment and medical imaging. Dr. Sureshi Rai; I first know him from the Wavelet class; I enjoyed his class while leading us going beyond symbols. Dr. Evangelos Triantaphyllou; a nice and understanding person (‘sas efharisto’); he keeps down your nervous, so the defense room gets cool down; but “make sure it is hundred percent correct and in perfect shape”.

And, with this dissertation accomplished, I am proud of trying to be in the chain of great people who dedicate their means to save life.

# TABLE OF CONTENTS

# ABSTRACT

Lung cancer is one of the most lethal cancer types. Research in computer aided detection (CAD) and diagnosis for lung cancer aims at providing effective tools to assist physicians in cancer diagnosis and treatment to save lives.

In this dissertation, we focus on developing a CAD framework for automated lung cancer nodule detection from 3D lung computed tomography (CT) images. Nodule detection is a challenging task that no machine intelligence can surpass human capability to date. In contrast, human recognition power is limited by vision capacity and may suffer from work overload and fatigue, whereas automated nodule detection systems can complement expert's efforts to achieve better detection performance.

The proposed CAD framework encompasses several desirable properties such as mimicking physicians by means of geometric multi-perspective analysis, computational efficiency, and the most importantly producing high performance in detection accuracy. As the central part of the framework, we develop a novel hierarchical modular decision engine implemented by Artificial Neural Networks. One advantage of this decision engine is that it supports the combination of spatial-level and feature-level information analysis in an efficient way. Our methodology overcomes some of the limitations of current lung nodule detection techniques by combining geometric multi-perspective analysis with global and local feature analysis. The proposed modular decision engine design is flexible to modifications in the decision modules; the engine structure can adopt the modifications without having to re-design the entire system. The engine can easily accommodate multi-learning scheme and parallel implementation so that each information type can be processed (in parallel) by the most adequate learning technique of its own. We have also developed a novel shape representation technique that is invariant under rigid-body transformation and we derived new features based on this shape representation for nodule detection.

We implemented a prototype nodule detection system as a demonstration of the proposed framework. Experiments have been conducted to assess the performance of the proposed methodologies using real-world lung CT data. Several performance measures for detection accuracy are used in the assessment. The results show that the decision engine is able to classify patterns efficiently with very good classification performance.

# CHAPTER 1. INTRODUCTION

## 1.1) Motivation

Lung cancer is one of the most lethal cancer types. It is reported in [Partain05] that it accounts for 32% and 25% of cancer deaths among men and women [Dood04] and causing 150,000 deaths a year [Li2003] in the United States, respectively. It is proved that early detection of this cancer type may increase the chance of surviving [Sluimer06, Dood04, Li2003]. It has been observed that radiologists may overlook some nodules due to heavy load of number of images and fatigue. It is also proven that a computer aided detection and diagnosis CADD system can provide a 'second opinion' for radiologists to increase their interpretation performance [Doi005].

[Factus] reported some facts about the lung cancer in the United States: "Approximately 175,000 people are diagnosed with lung cancer in the U.S. each year– about 82,000 women and nearly 93,000 men. Lung cancer kills more than 160,000 people annually – more people than breast, colon and prostate cancers combined. Lung cancer is responsible for more than a quarter of all cancer-related deaths every year. Smoking is the primary cause of lung cancer. Approximately 90 percent of lung cancer cases occur in people who are currently smoking or have previously smoked. Although the risk of developing lung cancer goes down with smoking cessation, a significant risk remains for 20 years or longer after quitting. Approximately 50 percent of all lung cancers (87,500) occur in people who have already quit smoking. Radon exposure is the second leading cause of lung cancer, and the leading cause of lung cancer among never-smokers. More people who have never smoked die from lung cancer than do people from AIDS or liver cancer or ovarian cancer. Risk factors for lung cancer other than those from smoking include lung scarring from tuberculosis, and occupational or environmental exposures to radon, second-hand smoke, radiation, asbestos, air pollution, arsenic and some organic chemicals. Only 16 percent of lung cancer patients are diagnosed before their disease has spread to other parts of their bodies, (e.g., regional lymph nodes and beyond), compared to more than 50 percent of breast cancer patients, and 90 percent of prostate cancer patients. Men's mortality (death) rates from lung cancer began declining over 20 years ago, while women's lung cancer mortality rates have been rising for decades and just recently began to stabilize. African Americans experience the highest incidence of lung cancer, and the highest death rate. Roughly, 85 percent of people diagnosed with lung cancer die within five years of their diagnosis, compared to 12 percent of breast cancer and 1 percent of prostate cancer patients. Less money is spent on lung cancer research compared to research on other cancers. In 2005, the National Cancer Institute estimated it spent only $1,708 per lung cancer death compared to $13,947 per breast cancer death, $10,214 per prostate cancer death and $4,655 per colorectal death."**.**

As seen from the facts stated above, invention of a novel methodology that can assist to detect a cancer would be very crucial. It has been experienced for a long time that a CADD system is one of the important tools used in medicine. Main goal of a CADD system is to give a second opinion to enhance the performance of radiologist while increasing the number of true positive detections and decreasing the number of missed nodules. A CADD system is mainly used to assist human interpreters to identify and characterize abnormalities automatically [Ko2004] "making lesions easier to detect and classify and potentially identify at an earlier stage" [Krup04]. Although automatic nodule detection systems may yield false outputs, [Ko2004] showed that image interpretation time significantly improved. Besides assisting in interpretation, a CADD system can provide a standard against both the inter- and intra-observer variations in identification as well [Ko2004] [Krup04]. The future of CADD system tends to "provide some possibility of being malignant vs. benign and perform tailored image processing to enhance the visibility of the lesion for the radiologist so that it can be readily perceived and interpreted" [Krup04]. Another advantage of using a CADD system is to reduce the workload of radiologist and consequently decreasing error as well [Arma04]. As stated in [Rubin00], with the advanced technology of CT scanning, 'data explosion' is a big challenge when we consider that a radiologist has to analyze more than 300 image sections per case. Explosion of image data affects human judgment due to physiological exhaustion like fatigue or distraction [Arma00].

The frontal-chest-radiograph as compared to CT scanning has some disadvantages of projectional imaging such as obscuration due to the overlaying structures and lack of discrimination between texture of normal and abnormal tissues, and insensitivity to early stages of cancer [Hans00]. On a research, it was found that the average size of missed peripheral cancers by radiologist is 0.3 cm for CT versus 1.3 cm for radiography [Arma00]. The spiral CT has an advantage over chest radiography as "shorter scanning time, lower radiation exposure, and increased diagnostic accuracy" [Kane96]. In spite of the advancement of CT scanning technology, missed nodules are still a challenge due to the structure of the lung anatomy. Among the pulmonary nodule types, pleural tumors, vascularized nodules, or those having the size of less than 5 mm are the mostly missed nodule types [Ko2004]. Another challenge in automatic reading of a CT section image is related to the partial volume effect and anisotropic volume reconstruction [Kuhn06].

## 1.2) Problem Definition and the Proposed Framework

Nodule detection techniques can be categorized with respect to their spatial analysis dimension such as 2D or 3D detection and their analysis granularity that can be global or local. It is obvious that high-dimensional analysis is superior to lower ones as regard to detection accuracy; whereas it demands more computational resources. Global analysis approaches enable to characterize the whole volume of interest (VOI) with higher-level abstraction, which may be desirable in terms of computational resource allocation; on the other hand, discriminative power would be so restrictive. In contrast, local analysis approaches can provide more descriptive features while leading to the curse of dimensionality that may require advanced processing techniques; even in some cases, the dimensionality may gets too large and making decision on the class of a given object can be an NP-complete problem.

As an important fact that human intelligence is the best nodule detector and yet there is no machine intelligence that can compete with this human detection capability. Whereas human visual recognition power is limited by eye vision capacity, on which the machine can be superior. Therefore, a comprehensive nodule detection technique should be able to combine higher spatial dimensional analysis with global and local feature level analysis while demanding less computational resources and high performance detection accuracy, and be able to mimic human intelligence. In addition, the detection system should be flexible so that different classification approaches can be used together and modification of its units should not require redesigning the entire system.

According to our extensive literature review about pulmonary nodule detection from 3D CT images, the current approaches analyze a VOI for suspicious objects by 1) searching through transverse plane or 2) performing segmentation. The first group approaches aim either to find the most descriptive section or to model the VOI based on some spatial assumptions before extracting characteristic features; the second group approaches perform, first, a 3D segmentation and then extract features for pattern recognition. On the other hand, they perform feature level analysis in a non-categorical way; so the size of a feature vector has to face curse of dimensionality. Another drawback of the current approaches, which employ 3D analysis, is that they consider entire VOI to produce some global feature vectors, which have less capability of characterizing complicated lung volumes.

The objective of this dissertation is to propose a novel CAD framework that can encompass properties of the comprehensive nodule detection framework described above. Accordingly as a part of the framework, we proposed and implemented a novel modular nodule detection method and a novel content based shape representation from which new shape features are derived. The proposed framework is composed of integrated units that automate the whole process including data retrieval, pre-processing, feature extraction, training, testing and assessment for detection. An important aspect of the proposed decision engine is that its integrated units at the same layer can process independently; therefore, its structure is very well suited for parallel processing environment.

### 1.2.1) The Framework Proposed

The organization of the framework is depicted in Figure 1.1 and Figure 1.2. The data retrieval unit reads nodule location information and corresponding image series locations from the data archive for each specified patient. All volume VOIs are extracted for further processing. In the preprocessing unit, gray-level and scale normalization are applied; after that, coronal and sagittal volumes are computed from sections; finally, sections that are not adequate for feature extraction are replaced with their nearest ‘usable’ sections. The feature extraction unit produces feature vectors for each section of planes. At the feature set preparation stage, extracted features are organized in such a way that their spatial order preserved per each image series; this special organizing is crucial to realize modularity. The next step is the main unit, the Decision Engine, where the classification is performed. All of the outputs from each unit are archived to be able use later, e.g. for testing and assessment purpose. Some units can allow a user interaction to change the parameters. The whole system is designed in such a way that ‘once the start button is pushed on this side, you can get the all results at the other side’; so that experiments can be conducted in an organized and efficient way.

The overall scenario is as such that a set of CT slices from a VOI and candidate nodule region of interests (ROIs) are given, these slices are preprocessed to handle anisotropy and scalability problems, and to normalize their gray level range. After the preprocessing, a VOI is obtained from the raw data by selecting/replacing some regions from the VOI; following that, feature vectors are obtained from this VOI using feature extraction tools that we developed. After extracting all feature vectors for each slice, they are organized in a special way for modular processing. At the final stage, the modular Decision Engine processes these feature sets to produce the final classification decision. The whole system is implemented in the MATLAB environment. A user interface is developed to analyze image series, to perform some image processing, and to extract VOIs.

s

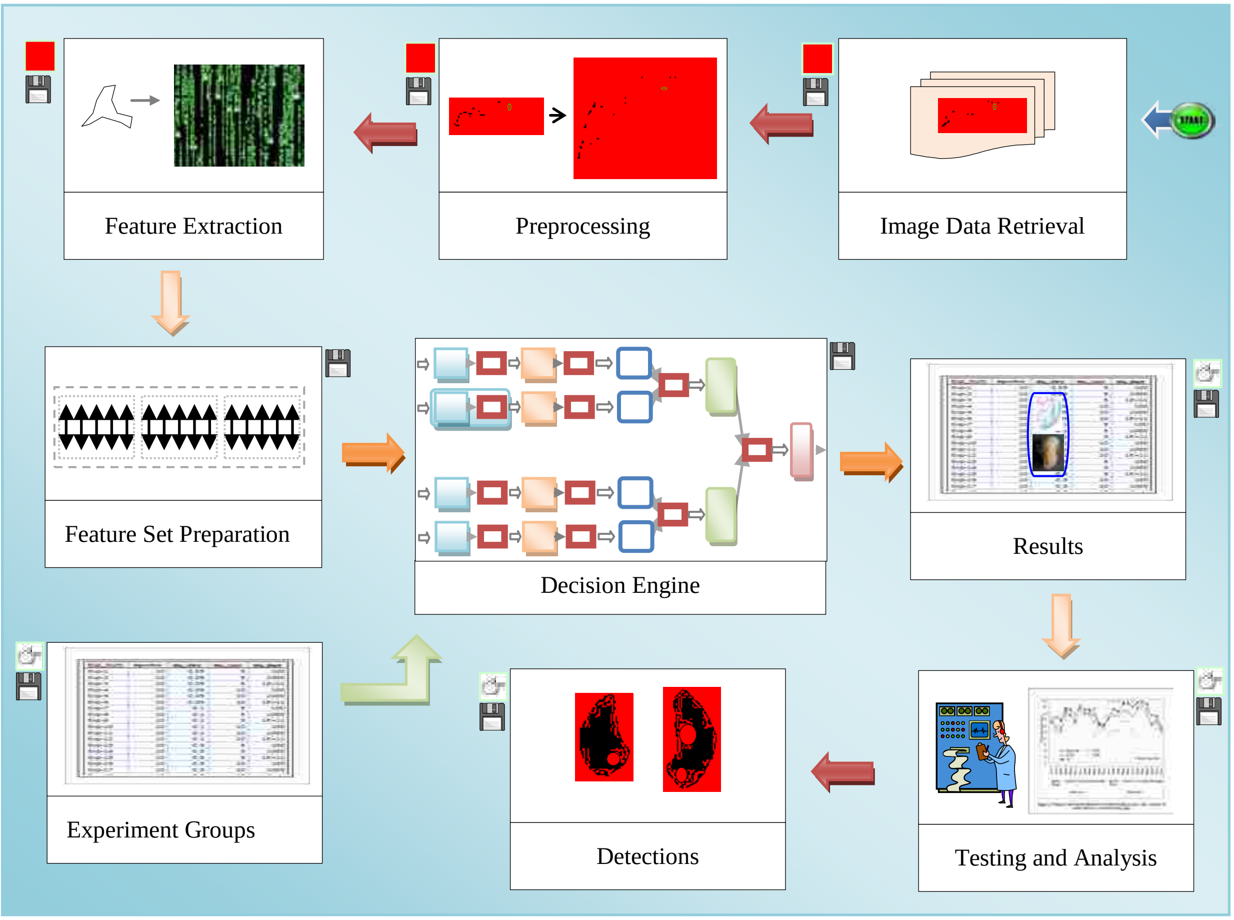


Figure 1.1 The proposed framework

The rest of the dissertation is organized as follows: In the second Chapter, background information is presented; the Section 2.1 is about medical imaging and CT, the Section 2.2 is about pulmonary anatomy, lung cancer, tumor and image characteristics of nodules, the Section 2.3 is about pattern recognition system,  basic structure of an Artificial Neural Networks (ANN), the Section 2.4 explains basics of Backpropagation learning and two versions of this method that is utilized in this study are explained briefly, and an extensive literature review is given in Section 2.5. In the third Chapter, content-based image representation is explained in detail; first, the concept of image content and principal of image representation is explained, and then, a novel shape representation is introduced in Section 3.3. In Section

3.4, we explained the feature types utilized in this research and introduced a new shape feature and two circularity features obtained from our canonical shape representation. The fourth Chapter is about data preparation; we explained how the raw data is collected and processed to prepare feature vector sets for training and testing. In addition, in this chapter, an auto-polygonalization algorithm is proposed. The fifth chapter covers our one of the main contribution ‘modular decision engine model’. The chapter six introduces performance assessment measures. In the Chapter 7, we discuss results obtained from implementation of the model and provide a comparison table with the relevant nodule detection studies. Finally, at Chapter 8, the whole system is summarized and possible future directions are listed

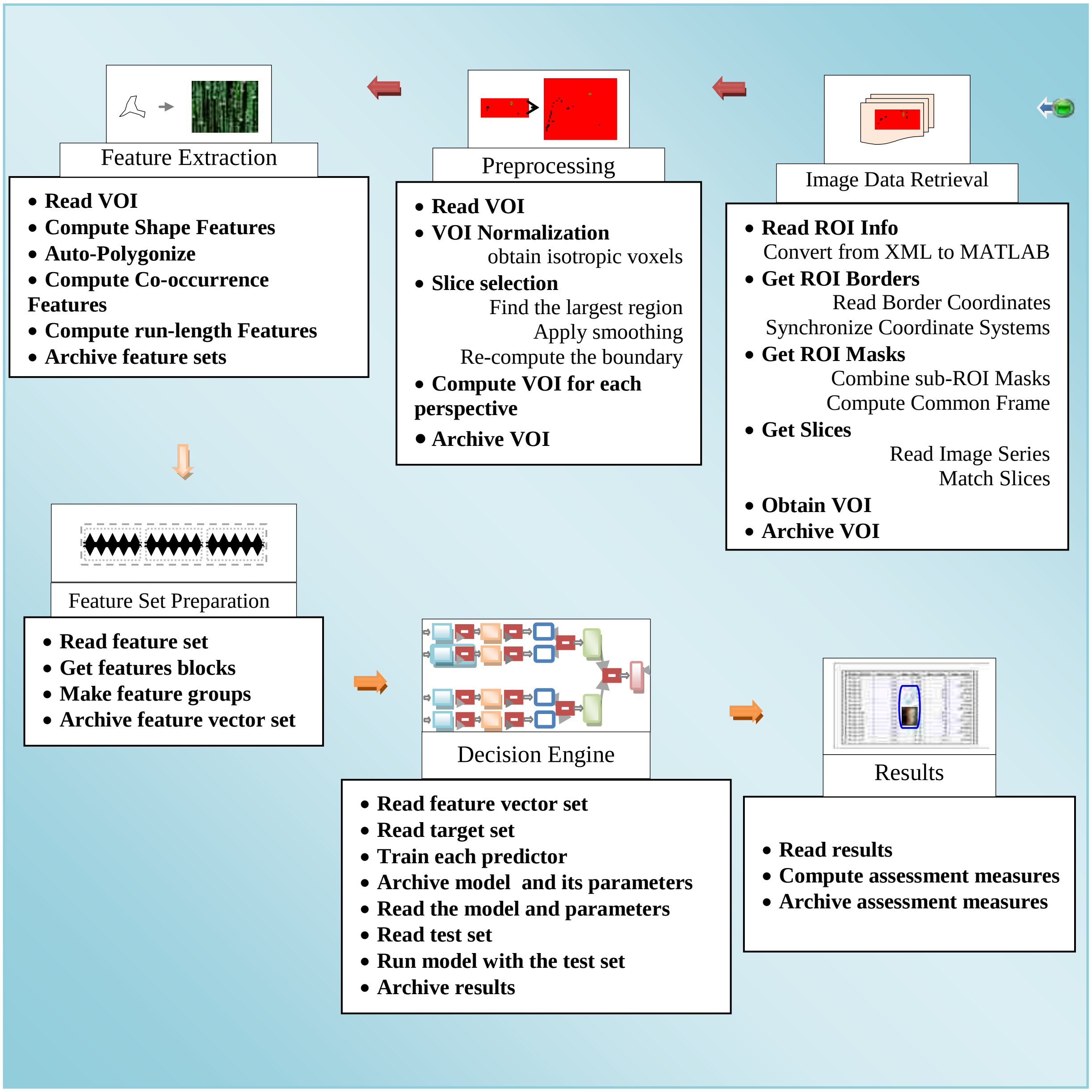


Figure 1.2 The proposed framework (with detail internal processes)

# CHAPTER 2. BACKGROUND

A CADD system provides a ‘second opinion’ [Partain05] to assist radiologist in reading and understanding of images with a certain accuracy and efficiency. Therefore, the goal of a CADD system is to provide necessary tools that are used to improve radiologic interpretation of lesions by efficient detection techniques to achieve more effective treatment. Output of a CADD system should yield a high true-positive rate while producing less number of false-positives as much as possible and it should be helpful to reduce intra-/inter-observations [Krup04]. As seen in the Figure 2.1, A CADD system is composed of following tasks or of sub-systems [Partain05]: Data acquisition, preprocessing, feature extraction, detection, diagnosis, retrieval of similar images in the database, and assessment of outputs.

A more enhanced CADD system is given in Figure 2.2. Phases of the CAD framework are retrieval of images, preprocessing, lung region extraction and lung boundary tracing, candidate search, feature extraction, and nodule detection and diagnosis. The flow of the CAD system is as follows: First, pulmonary CT images are entered to the system. Second, anisotropic voxels are preprocessed by an interpolation technique. After that, the lung region is extracted to reduce the search time. Once the lung region is determined, candidate VOIs are searched within the lung volume. Then, features that describe an image object in the VOI are extracted from each slice of three orthogonal planes. The classifier utilizes these features to make decision about the class of the candidate imject. During the process, an interaction is available in extraction, search and database units. Extracted lung lobes, features, and candidate imjects can be archived to the database system to be used later. Rules that control the decision engine in detection unit can be modified based on new experiences.

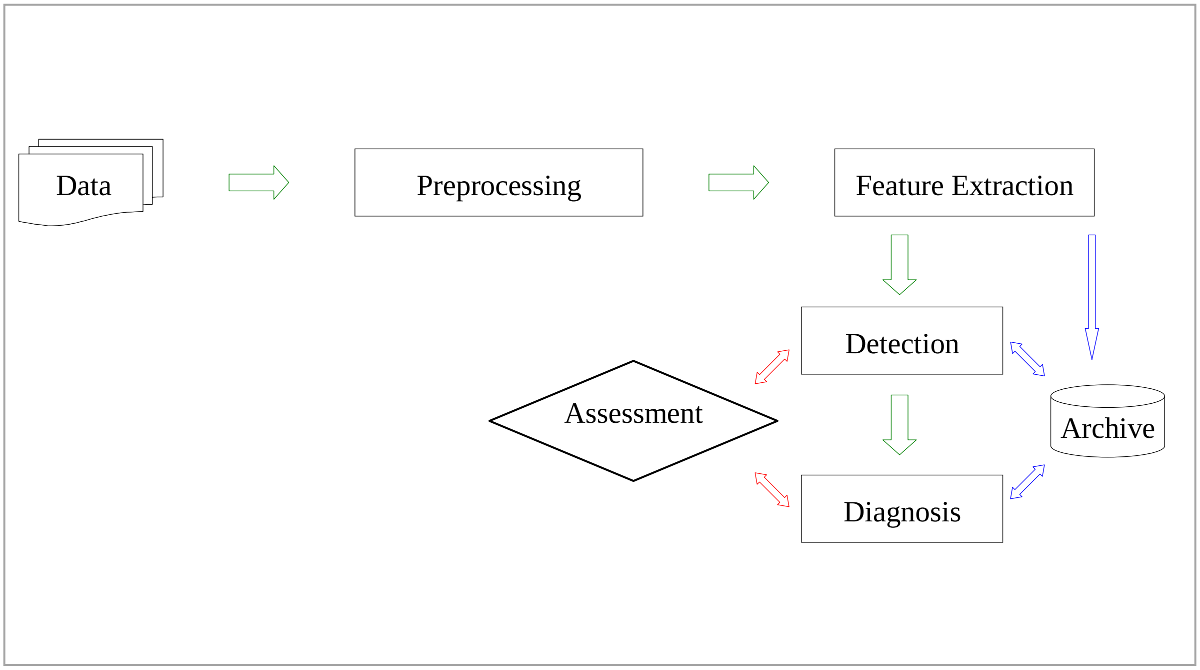


Figure 2.1 A typical CADD system

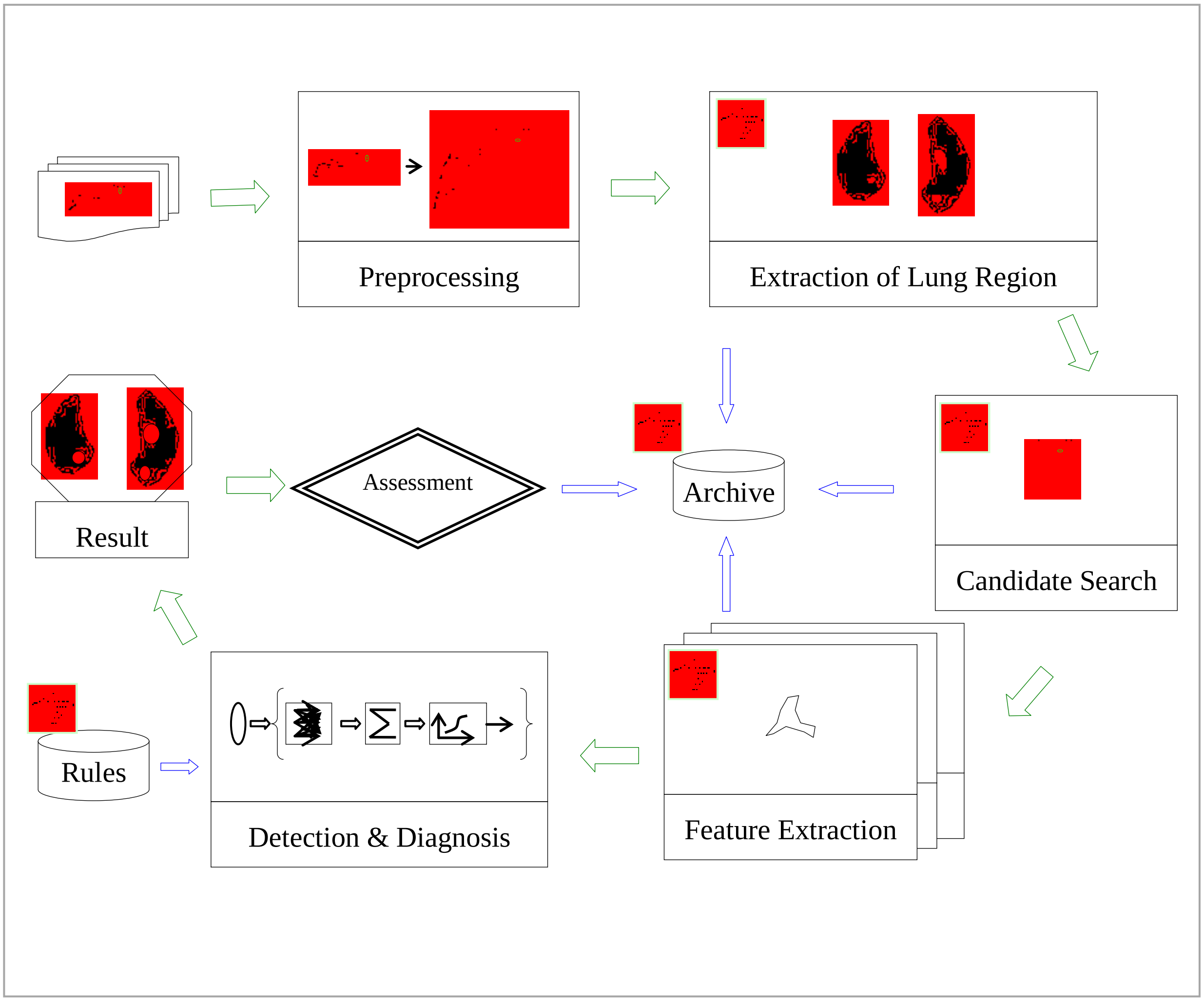


Figure 2.2 An enhanced CADD system

## 2.1) Medical Imaging

Medical imaging utilizes some forms of energy interacting with tissues [Bush01]. The spectrum of the energy used in medical imaging can be in visible range as in skin-photography or light-microscopy, or can be in X-rays range as used in radiology. The computed tomography imaging system is explained in the sequel.

### 2.1.1) Computed Tomography

The term *tomography* is assembled from *graph* and *tomo* (slice) meaning picture of a slice [Bush01]. A tomographic image is reconstructed by a computer system based on the energy acquired from different views. The Computed Tomography utilizes X-ray spectrum energy. This energy is directed to penetrate tissues. The amount of interaction — in the form of attenuation, scattering or absorption — during the penetration is measured to create image of the body. The variation of interaction for each tissue type enables to create a contrast. If all tissues react with the energy in the same way, there will be no contrast, hence no meaningful image.

A CT imaging system is composed of several units such as energy source unit, energy management unit, detection unit, movement control unit, and reconstruction unit. The X-ray energy is produced and directed by a beam in the form of rays. A ‘ray’ is a single transmission measurement detected by a single detector at a moment. Several rays can be sent from a ‘parallel beam’ source or from a ‘fan beam’ source in different ‘projections’. ‘View’ or ‘projection’ is the orientation through which several rays are radiated. Modern CT scanners employ fan-beam geometry. The goal of energy source and management unit is to utilize maximum amount of measurement from different projections accurately while keeping the amount of energy dose at a minimum level in a limited time. A typical example, 800 rays taken at 1,000 different projections will rise about 800, 000 single transmission measurements [Bush01 p. 329].

The detection unit is constructed from series of arrays of detector hardware where the energy leaving the body is acquired. The detector arrays can be in different forms such as single arrays and multiple arrays. The state-of-art technology utilizes the multiple detector array systems. A detector unit can be Xenon detector or a solid-state detector, which is mostly used. The X-rays incident on a scintillator that is the top part of a detector are converted to visible light spectrum. The light emitted by scintillator is detected by a photo-detector. The photo-detector converts the light energy into electrical energy proportional to the light intensity.

Movement and position of the table, beam or detector characterize the technology of the imaging system. For example, a helical CT scanner enables the table movement while rotating continuously beam and detector array chamber. The data acquisition time and accuracy is affected by movement and position parameters.

Reconstruction unit utilizes the raw data obtained from detection unit to create a 2D image of the volume scanned using a reconstruction algorithm. The most widely used reconstruction algorithm is ‘filtered backpropagation’ algorithm. Data acquired by detectors from different orientations is preprocessed to decrease the effect of blurring by convolving with a ‘$1/r$’ filter such as ramp filter. In addition, some other corrections on the raw data such as related to geometric discrepancies are accomplished. After the preprocessing, an attenuation coefficient is computed for each ray. A system of unknowns of voxel values is solved using these set of coefficients.

Attenuation of photons is proportional to tissue structure (soft or bone-type), photon energy level (typically <26 keV for low photon energies), and atomic number of the absorber (e.g. soft tissues have low atomic number). Some of photons are removed due to the attenuation. A CT scanner measures the ‘linear attenuation coefficient’ for each ray. The linear attenuation coefficient is “the fraction of photons removed from a mono-energetic beam of X- or Gamma rays per unit thickness of material” [Bush01 p. 45]. It is unit is mm$^{-1}$. The relation between number of photons removed $n$ and the linear attenuation coefficient $\mu$ for a ray traversing a small distance $\Delta x$ with some incident photons of $N$ can be given by equation (2.1) in case of a linear relation. This equation tells that, for each 1,000 incident photons, about 16 photons are removed from the beam traversing a soft tissue of $\mu$ = 0.016 mm$^{-1}$ (under 100 keV) passing through a thickness of 1 mm.

$$n = N\,\Delta x\,\mu \tag{2.1}$$

On the other hand, attenuation is not linearly proportional to the thickness. Equation (2.2) defines the non-linear relation between the number of incident photons $N_0$ and the number of photons $N$ succeeded to pass a distance $x$ through a tissue of the coefficient $\mu$. Considering the example given above, $N \approx 984$ and $n \approx 16$; but if $\Delta$x = 60 mm, then $N \approx 382$ and $n \approx 617$ not $n \approx 960$ (in case of a linear relation).

$$N = N_0\, e^{-\mu\, x} \tag{2.2}$$

The linear attenuation coefficient is combination of linear coefficient of different type of linear attenuation coefficients such as Rayleigh, photoelectric effect, Compton scatter, and pair production.

A contrast on a tomographic image is due to different interaction of tissues with X-rays passing through. The linear attenuation coefficient is computed using equation (2.3) after measuring number of incident photons on a detector element for a traveling distance of $x$. If we sub-divide the total distance $x$ traveled into $k$ small intervals and considering the linear attenuation coefficient is constant for each sub-interval, then equation (2.4) shows the relation between attenuation coefficients of each individual sub-intervals and the measured linear attenuation coefficient [Bush01 p. 350].

$$\mu = \frac{\ln(N_0/N)}{x} \tag{2.3}$$

$$\mu = \frac{\sum_{i=1}^{k} \mu_i/k}{x} \tag{2.4}$$

After computation of the linear attenuation coefficients for each ray, system of unknowns of voxel values is solved using these set of coefficients and voxel values. These voxel values are represented by high-precision floating point number, are transformed to Hounsfield Unit (HU) for display purpose. The CT value of each voxel is computed by equation (2.5). A CT value, which is given in Hounsfield Unit, is an integer value due to truncation. Some typical CT values are 0 HU for water, -1000 for air, [-300, +100] for soft tissues, and up to 3000 for bones.

$$CT(x,y) = 1000\,\frac{\mu(x,y) - \mu_{\text{water}}}{\mu_{\text{water}}} \tag{2.5}$$

CT values are typically stored in 12-bit depth; hence, 4096 different gray levels can be obtained. The range of a typical CT image is [–1000, +3000].

## 2.2) Pulmonary Anatomy and Lung Cancer

Pulmonary (pulmo- is from Latin pulmonarius ("of the lungs") [Wilung]) anatomy studies structure of lung. In the following sections, anatomy of pulmonary region and lung cancer will be briefly explained.

Lung is the essential organ of the respiratory system. “Its principal function is to transport oxygen from the atmosphere into the bloodstream, and to excrete carbon dioxide from the bloodstream into the atmosphere. This exchange of gases is accomplished in the mosaic of specialized cells that form millions of tiny, exceptionally thin-walled air sacs called alveoli” [Wilung]. The lung anatomy is shown in Figure 2.3.

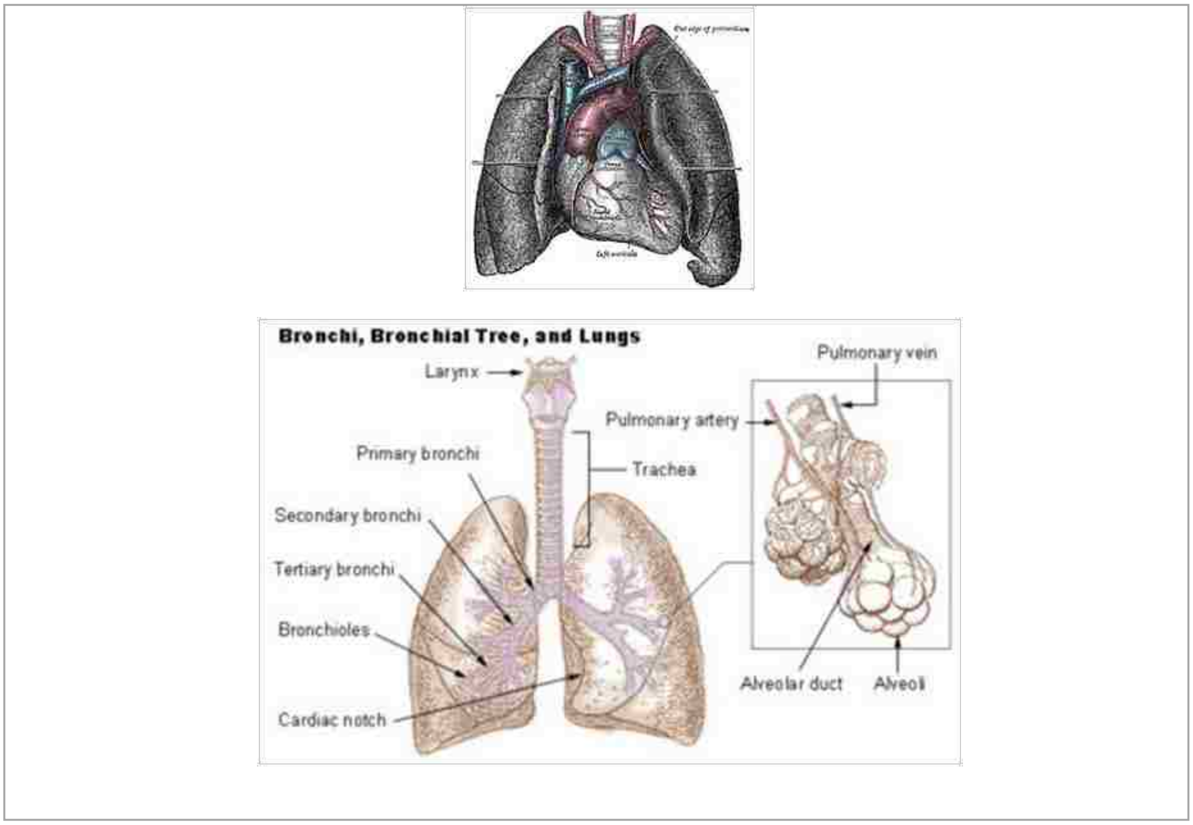


Figure 2.3 Lung anatomy [Wilung]

In humans, the trachea bifurcates and produces two main bronchi. The bronchi give rise to bronchioles after multiple divisions. The bronchial tree ends with alveolar sacks. Clusters of alveoli made up alveolar sacks. Blood vessels tightly wrap the individual alveoli [Wilung]. “Human lungs are located in two cavities on either side of the heart. Though similar in appearance, the two are not identical. Both are separated into lobes, with three lobes on the right and two on the left. The lobes are further divided into lobules, hexagonal divisions of the lungs that are the smallest subdivision visible to the naked eye. The connective tissue that divides lobules is often blackened in smokers and city dwellers. The medial border of the right lung is nearly vertical, while the left lung contains a cardiac notch. The cardiac notch is a concave impression molded to accommodate the shape of the heart. Lungs are to a certain extent 'overbuilt' and have a tremendous reserve volume as compared to the oxygen exchange requirements when at rest. This is the reason that individuals can smoke for years without having a noticeable decrease in lung function while still or moving slowly; in situations like these only a small portion of the lungs are actually perfused with blood for gas exchange. As oxygen requirements increase due to exercise, a greater volume of the lungs is perfused, allowing the body to match its $CO_2/O_2$ exchange requirements." [Wilung].

"Lung cancer is the malignant transformation and expansion of lung tissue, and is the most lethal of all cancers worldwide, responsible for 1.2 million deaths annually. It is caused predominantly by cigarette smoking, and predominantly affected men, but with increased smoking among women, it is now the leading cause of death due to cancer in women. However, some people who have never smoked still get lung cancer." [Wilucan].

"Cancers that begin in the lungs are divided into two major types, *non-small cell lung cancer* and *small cell lung cancer,* depending on how the cells look under a microscope. Each type of lung cancer grows and spreads in different ways and is treated differently. Nonsmall cell lung cancer is more common than small cell lung cancer, and it generally grows and spreads more slowly. There are three main types of non-small cell lung cancer. They are named for the type of cells in which the cancer develops: *squamous cell carcinoma* (also called *epidermoid carcinoma*), *adenocarcinoma*, and *large cell carcinoma*. Small cell lung cancer, sometimes called *oat cell cancer*, is less common than non-small cell lung cancer. This type of lung cancer grows more quickly and is more likely to spread to other organs in the body." [Cangov].

A *tumor* is "primarily used to denote abnormal growth of tissue" [Witumor]. A tumor can be invasive or non-invasive. *Invasive tumor* is an abnormal tissue growth that spreads (invades) to the surrounding healthy tissue (usually a malignant one). *Non-Invasive* tumor is an abnormal tissue growth that does not spread to the surrounding healthy tissue. "*Cancer* is a class of diseases or disorders characterized by uncontrolled division of cells and the ability of these cells to invade other tissues, either by direct growth into adjacent tissue through invasion or by implantation into distant sites by metastasis." [WikiCan].

Solitary pulmonary nodules (SPN) are defined as 'focal, round, or oval areas of increased opacity in the lung that measures less than 3 cm in diameter" [Eras00, Worman04]. In [Ge2005], it is defined as "single round intra-parenchymal opacity, at least moderately well-marginated and less than 3 cm in maximum diameter". "Lesions measuring greater than 3 cm are classified as masses" [Buny06]

Pulmonary nodules can be classified into four groups based on their locations [Kosti03]: 1) Well-circumscribed: The nodule is located centrally in the lung, without significant connections to vasculature. 2) Vascularized: The nodule is located centrally in the lung, but has significant vascularization (connections to neighboring vessels). 3) Pleural tail: The nodule is near the pleural surface, connected by a thin structure ("pleural tail"). 4) Juxtapleural: A significant proportion of the nodule periphery is connected to the pleural surface.

### 2.2.1) Image Characteristics of Nodules

Basic image characteristics of a nodule can be determined by its morphological features and internal characteristics. In this section we briefly explain these characteristics.

Morphologic features**:** Size, margin, and contour; “nodule margins and contours can be classified as smooth, lobular, irregular, or speculated” [Eras00]. Assessment of margin characteristics may not be a valid discriminator [Worman04].

Internal Characteristics**:** Basic internal characteristics of a nodule image are attenuation features, type of cavity, and its calcification pattern [Eras00]. Nodules can be characterized by their CT values conveyed based on homogeneity, contrast enhanced-CT measure, CT number level, and opacity. CT values are similar for both tumor and vessels [Kanaza96]. CT values are relatively uniform for malignant nodules compared to the vessels. CT values are generally higher than that of vessels in vertical sections [Satoh99]. Homogenous attenuation is seen in both benign and malignant [Eras00, Worman04]. “Shadows contacting the lung wall are generally tumors” [Satoh99].

Level of CT numbers for lung structures: Air: -1000 HU, lung tissue: -910 to – 500, chest wall, blood, and bone: above -500 [Hu2001]. Intra-nodular fat: -40 to -120 HU [Eras00, Worman04].

Opacity: “Non-solid and part-solid nodules, also referred to as ground-glass opacities (GGO), are entities that have attracted increasing interest and systematic analysis over the past few years. They were initially observed in trials for early detection of lung cancer with low-dose CT and were found to represent lung cancer in a higher proportion of cases than solid nodules (malignancy in 34% of GGOs vs. 7% of solid nodules in one study). The distribution of histological findings in malignant GGOs is different from findings in solid nodules: GGOs frequently represent bronchioloalveolar carcinoma or adenocarcinoma with bronchioloalveolar features, whereas histological analysis of solid nodules more often reveals others subtypes of adenocarcinoma, squamous cell carcinoma and small cell lung cancer” [Worman04].

Cavitation: “Benign cavitary nodules generally have smooth, thin walls” while “malignant nodules typically have thick, irregular walls” [Eras00].

Wall thickness: In general > 16 mm is a sign of malignancy, < 4mm is a sign of benign [Eras00, Tan2003].

Calcification Patterns: Attenuation value of 200 HU can be used to discriminate calcified and non-calcified nodule. [Eras00]. Eccentric or stippled form of calcifications is indeterminist [Tan2003].

## 2.3) Pattern Recognition

A pattern is a signal whose elements have some common contextual characteristics. A contextual characteristic for a pattern in an image can be defined based on a geometric, a topological, or photometric structures. An ‘imject’ (image object or object in an image) can be defined by means of patterns conveyed by the image object such as an imject of a vessel can be defined by a gray-level profile that is perpendicular to its major axis. Patterns can be grouped into two main categories namely concrete and abstract. A concrete pattern is based upon physical measurements over spatial and temporal items such as fingerprints, weather map waveforms, and an abstract pattern is based upon "attribute of concepts and mental models" [Looney97, Jesa04]. In this dissertation, 'pattern' refers to 'concrete pattern'.

Pattern Recognition, generalization, identification, and classification are closely related concepts. 'Pattern Recognition' is making a decision about class of an object by assigning its class to a known population [Looney97]. We can look at pattern recognition as an 'inverse problem', which is one-to-many if the forward problem is many-to-one and it can be solved using a mixture model [Bish06]. 'Identification' is "the recognition of an object as a unique singleton class" [Looney97]. In 'classification' process, the object is assigned to a class specified based on its features likelihood. A learning system reaches its entropic stable state when the system 'learns' the class of a pattern [Looney97]. 'Generalization' is one of the goal/aspects of a pattern recognition task [Bish06]. Recognition can be done by a 'supervised' or an 'unsupervised' learning approach. A supervised approach aims to assign a pattern to a predefined class while an unsupervised approach aims to group patterns according to a similarity metric. A recognition process can utilize statistical (or decision theoretic) or syntactic (or structural) approaches [Looney97 p. 8]. "'Statistical pattern recognition' is based on statistical characterizations of patterns, assuming that the patterns are generated by a probabilistic system. 'Structural pattern recognition' is based on the structural interrelationships of features." [Foldoc]. 'Pattern Recognition System' is composed of data acquisition, preprocessing, pattern extraction, feature extraction, classification (or identification), and post-processing units [LeonC98, Jesa04 p. 2, Looney97 p. 5 Figure 0.1, Duda01].

Classification approaches can be dichotomized in two main groups: Statistical (decision theoretic) approaches and syntactical (structural) approaches [LeonC98 p. 21]. In our research, we will develop an ANN-based recognition method, which is a structural approach. Designing a novel classifier to cope with some variations due to noise and data acquisition system [Duda01, p. 12] is a challenging problem of a pattern recognition system. Classification (from the decision theory point of view) can be done by minimizing the misclassification rate, maximizing the true classification rate, or further minimizing the expected loss [Bish06 p. 38-48].

### 2.3.1) Pattern Extraction

Patterns in an image can be extracted by a segmentation method, a filter-based technique, or a model-matching approach. Segmentation in Computer Vision is assigning pixels to some classes such that each segment has the maximal region. Filter-based techniques utilize some spatial filters, which are traversed over the image to extraction a pattern according to filter specification. Sobel's mask [Ziou97] and Canny edge detector [Sonka98] are some examples of spatial filters. In model-matching approaches, patterns are defined by model parameters. Model parameters can be statistical distribution of some features such as Gaussian distribution of gray-levels within a window or can be some shape features such as polygonal shape of an imject's class. Segmentation methods and model-matching approaches use some similarity metrics to get the optimum result. Some examples of similarity metrics are Euclidian distance, Mahalanobis distance, Minkowski distance, Kullback-Leibler distance.

Patterns like 'blob' are helpful in detection of nodules due to their spatial similarity in 2D space. Blob can be defined as "a connected region that is either significantly brighter or significantly darker than its neighborhood. It should have a sufficiently large area and be stable over some sufficiently large interval in scale-space" [Lindeb91].

### 2.3.2) Pattern Classification Using ANN

Artificial neural network (ANN) is a classification technique that applies "biological concepts to machines to recognize patterns" [Jesa04 p. 3]. Its root goes to the article by McCulloch and Pitts in 1943 [McCul43]. Although it emerged by inspirations from biological models, there is a strong relation between ANN and the statistical discriminant approaches [Duda01 p. 21, 33, Bish06, Lewis03]. ANN can be used to compute posterior probabilities, it can also be utilized for linear discriminant analysis as well as for quadratic discriminant analysis [Zhang00]. In addition, it has a power of flexibility employing non-linear mapping and it is a "data-driven self-adaptive method" without imposing a specific function or distribution [Zhang00]. An extensive comparison study reported in [Zhang00] concluded that there is no best classifier that works for all data sets. Main advantages of using ANN are adaptive-learning, self-organization, fault tolerance capabilities, generalization power, massive parallelism, and distributed representation [Jesa04 p. 4, Jain96, Schalk97]. Challenges in using ANN include the over-fitting problem, difficulty in prediction of generalization, and lack of specific design guidelines that can be applied in general [Schalk97 p.10]

### 2.3.3) A Basic Structure of ANN

A neural network model is a nonlinear mapping of input space $\{x_i\}$ to output space $\{y_j\}$ controlled by some adjustable parameters $\{w_k\}$ [Bish06 p. 228]. It implements linear discriminants while learning nonlinear mapping functions [Duda01 p. 283]. It is a heuristic and adaptive pattern recognition approach. It is composed of three layers namely input layer, hidden layer(s), output layer. Every layer is composed of nodes namely input nodes, hidden nodes, and output nodes. Each node has two units: First unit is to compute activation and the second unit is to transform activation. The activation, sometimes called 'net', is linear combination of weighted inputs as defined in equation (2.6). The second unit is to transform the activation to a value between two saturation values using, in general, a non-linear function. A threshold also called bias can be used to adjust the location of the decision plane. The bias can be introduced as a part of the input vector as well. Figure 2.4 shows a typical ANN structure [Lewis03 p. 4]. A basic structure of a node (or a neural unit) which is also called perceptron is shown in Figure 2.5. Hidden nodes are used to transform non-linearly separable problems into linear one [Duda01 p. 299].

$$net = f(\mathbf{w}^{\mathrm{T}}\mathbf{x}) \tag{2.6}$$

### 2.3.4) ANN Topologies

ANN techniques can be classified as feed-forward (acyclic) or recurrent (cyclic) based on their connection topologies. Examples of the feed-forward topologies are Single-layer Perceptron, Multi-layer Perceptron (MLP), Adaptive Linear Element (ADALINE), Multiple Adaline (MADALINE), Radial Basis Function (RBF) network, and examples of the recurrent topologies are Kohonen Self-organizing Network, competitive networks, Hopfield network, Adaptive Resonance Theory networks (ART) [Jain96]. Another taxonomy of ANNs can be done based on the learning scheme they used; these are supervised or unsupervised learning schemes. In supervised learning, a corresponding target is specified for each training sample whereas in unsupervised learning samples are used without any target or 'teacher' to adjust learning weights. Back-propagation, RBF networks are typical examples of supervised approaches and Hebbian learning and competitive learning can be listed under unsupervised approaches [Kung93].

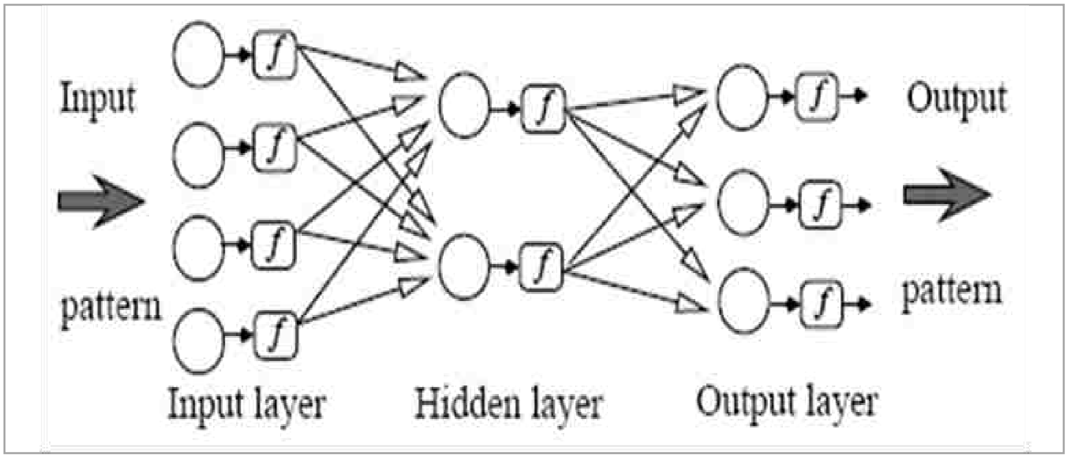


Figure 2.4 A basic ANN structure [Lewis03 p. 4]

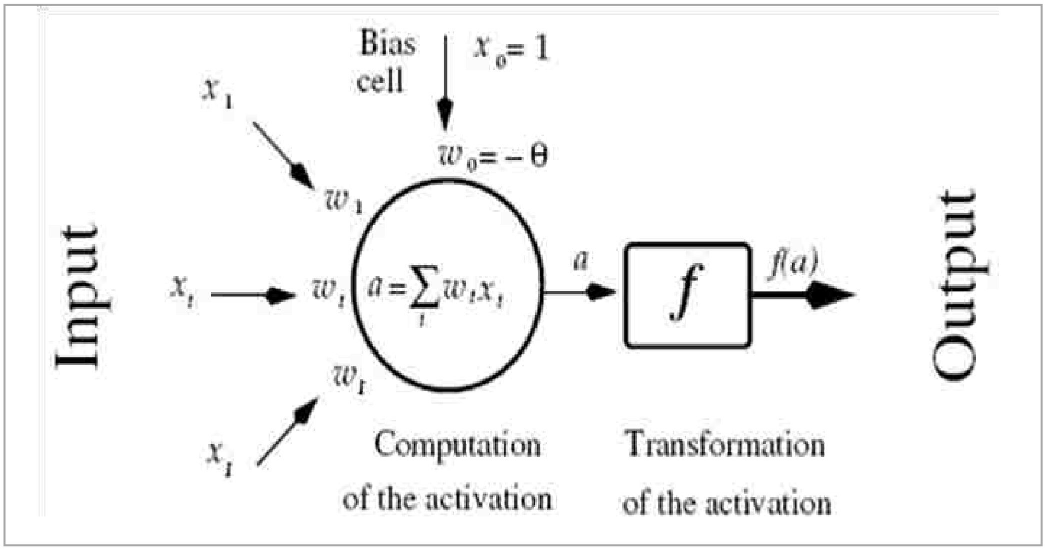


Figure 2.5 A basic neural unit [Lewis03 p. 4]

## 2.3.5) Training of ANN

There are three training protocols: Stochastic, batch, and on-line [Duda01 pp. 293, 316]. In stochastic training, patterns are selected randomly from a training set and weights are updated for each sample. In batch training, weights are updated after introducing all samples from a training set. In on-line training, samples are used only once and weights are updated as in the stochastic learning. In general, the number of training samples per class should be at least 5-10 times the dimensionality [LeonC98 p. 7]. In the sequel, we explain some basic parameters used in training of an ANN.

Activation function: Choice of activation function may differ depending on the nature of the problem. Linear activation functions are used for regression-type problems; logistic sigmoid and cross-entropy functions are used for binary classification [Bish06 p. 236]. If there is prior information about the distribution of the sample space, then an appropriate function e.g. a Gaussian function can be used. In case of back-propagation-based learning, an activation function should have following properties: 1) Non-linearity to utilize the computational power 2) It must saturate so output will be bounded 3) Continuity and smoothness to satisfy existence of derivative condition, and not as an essential but as a helpful property 4) Monotonicity not to introduce additional local minima in the error function, and 5) Linearity for small value of the activation [Duda01 p. 307].

Initializing Weights: Initialization of an ANN is important for the convergence. Finding the best parameter set of ANN for a given training data is an NP-complete problem [Leon98]. A poor weight initialization may yield the network to trap into a local minimum [Leon98 p. 88]. Weights should be initialized in such a way that a uniform learning, which requires adaption of all weights to be completed about the same time, can be achieved. A uniform weight update, that is while some weights do not converge faster than others, can be achieved selecting them from a uniform distribution of an interval [ $-\tilde{\omega},\tilde{\omega}$ ], and for a *d* dimensional input space we can chose $\tilde{\omega}=1/\sqrt{d}$ to make activation in the range of [-1, 1] [Duda01 p. 311]. Another approach to initialize weights is step-wise regression analysis by means of orthogonal least square or maximum covariance method [Leon98 p. 90-93]. Elimination of weights also can be done using second-order methods in network design [Duda01 p. 318]. A symmetric weight initialization in a network layer may cause to trap into a saddle point in case the initial weight and bias values are set to zero. In addition, large initial weight may produce a large output at the saturation region of the transfer function and consequently the performance function surface will produce some flatten regions away from the global minimum.

Learning Rate: The optimal learning rate is defined as the rate at which learning is achieved at one step. Learning rate is used to adjust converging rate. Accuracy of the final result is strongly affected by learning rate chosen [Duda01 p. 312].

Momentum: In gradient-based learning scheme, oscillation around a local minimum can be avoided by adding a fraction of change in the weight (obtained in the previous update step) so that it will accelerate the weight update step while descending down the hill, and will decrease update step size if sign of the error rate changes. This mechanism assists to escape from a local minimum. It is used to speed up learning [Schalk97 p. 164].

Hidden Layers: Hidden layers are employed to increase the dimensionality of mapping from input to output to have a better approximation [Bish06 p. 256]. On the other hand, increasing number of hidden layers may cause possibility of trapping into local minima [Duda01 p. 317-8]. Each node in a hidden layer can be thought as a (e.g. logistic) model that maps inputs to a class based on its parameters. Number of hidden layers can be determined by employing step-wise regression analysis with forward or backward elimination approach [Leon98 p. 90-93].

Matching Error (Cost) Function: Depending on the nature of the problem, the cost function may differ. Sum-of-squares [Bish06 p. 233] defined in equation (2.7) can be used for linear regression-type problems, cross-entropy error function defined in equation (2.8) can be used for binary classifications and multi-class classifications [Bish06 p. 236] or Minkowski error function [Duda01 p. 318] defined in equation (2.9) to adjust the locality of the classifier.

$$J_{\mathrm{ss}} = \sum_{m=1}^{n} \sum_{k=1}^{c} [z_{mk}(\mathbf{x}) - t_{mk}(\mathbf{x})]^2 \tag{2.7}$$

$$J_{\mathrm{ce}} = \sum_{m=1}^{n} \sum_{k=1}^{c} t_{mk}(\boldsymbol{x}) \ln\big(t_{mk}(\boldsymbol{x}) / z_{mk}(\mathbf{x})\big) \tag{2.8}$$

$$J_{\mathrm{Mink}} = \sum_{m=1}^{n} \sum_{k=1}^{c} |z_{mk}(\mathbf{x}) - t_{mk}(\mathbf{x})|^R \tag{2.9}$$

where $\mathbf{x}$ is input vector, $n$ and $c$ are number of samples and number of output neural units respectively, $z_{mk}$ is the output and $t_{mk}$ is the corresponding target for $m^{\text{th}}$ sample and $k^{\text{th}}$ output neural unit, respectively; $R$ adjust the influence of long tails in the distribution.

Over-fitting Problem in Training: The goal of designing a model is to obtain a good generalization over a sample space. How to achieve the best generalization and how to decide number of parameters is an important problem in design of ANN. If the number of weights are too many, then poor generalization (over-specialization); on the other hand, if they are too few, then poor learning performance (over-generalization) is obtained [Duda01 283]. Over-fitting is reduction in accuracy of classification over test data compared to the accuracy over training samples. When the number of the data points is more than the weights, an over-smoothing (or generalization) may occur. On the other hand, when the number of weights (mapping dimension) is larger than the number of exemplars, the network model will show ‘memorize’ effect as a consequences of over-specialization [Looney97 p. 307-308]. To avoid the over-fitting problem, several methods have been proposed such as weight decay (as a regularization term) and early-stopping.

## 2.4) Backpropagation Learning

Backpropagation learning algorithm was proposed by different scientist as a solution to non-linearly classification problems in 1980s. In this section we just introduce basics of the Backpropagation algorithm using equations in [Hagan95] and refer the reader to the reference for details. The Backpropagation is a version of the LMS (least mean squared) weight adjusting algorithm which uses the expected squared error as a performance measure; it differs from the LMS such that difference is the expected squared error is approximated by the squared error at an iteration *k*. In the next section the basic Backpropagation algorithm Steepest Descent and some variations of Backpropagation learning are explained briefly.

### 2.4.1) Steepest Descent Backpropagation Algorithm

Let *t*(*k*) and *a*(*k*) denote the target values vector and output of the activation function at an iteration *k*, respectively; the approximated error is given by

$$\hat{F} = \left(t(k) - a(k)\right)^T \left(t(k) - a(k)\right) = e^T(k)\, e(k) \tag{2.10}$$

So the steepest descent weight at an iteration *k* is computed by

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha \frac{\partial \hat{F}}{\partial w_{i,j}^m} \tag{2.11}$$

where $\alpha$ is the learning rate, *w* is the weight between *j*th input to the *i*th node at a layer *m*. The change in the error due to the change in weights can be calculated by

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = \frac{\partial \hat{F}}{\partial n_i^m} \frac{\partial n_i^m}{\partial w_{i,j}^m} = s_i^m \, \frac{\partial n_i^m}{\partial w_{i,j}^m}$$

$$s_i^m = \frac{\partial \hat{F}}{\partial n_i^m} \tag{2.12}$$

where the sensitivity $s$ denotes the change in $\hat{F}$ to the *i*th element of the net input at layer *m*, and the net input $n_i^m$ to the activation function of node *i* at layer *m* is given by

$$n_i^m = \sum_{j=1}^{S^{m-1}} w_{i,j}^m \, a_j^{m-1} + b_i^m \tag{2.13}$$

where *b* is the bias. Therefore

$$\frac{\partial n_i^m}{\partial w_{i,j}^m} = a_j^{m-1} \tag{2.14}$$

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = s_i^m \, a_j^{m-1} \tag{2.15}$$

Finally the approximate steepest descent equation will be

$$w_{i,j}^{m}(k+1) = w_{i,j}^{m}(k) - \alpha\, s_i^m\, a_j^{m-1} \tag{2.16}$$

In the matrix form, it is given by

$$\mathbf{W}^{m}(k+1) = \mathbf{W}^{m}(k) - \alpha\, \mathbf{s}^m \left(\mathbf{a}^{m-1}\right)^T \tag{2.17}$$

The sensitivity vector is computed using the recurrence relation given by

$$\begin{gathered}
\mathbf{s}^m = \frac{\partial \hat{F}}{\partial \mathbf{n}^m} = \left(\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m}\right)^T \frac{\partial \hat{F}}{\partial \mathbf{n}^{m+1}} = \dot{\mathbf{F}}(\mathbf{n}^m)\,(\mathbf{W}^{m+1})^T\, \frac{\partial \hat{F}}{\partial \mathbf{n}^{m+1}} \\
= \dot{\mathbf{F}}(\mathbf{n}^m)\,(\mathbf{W}^{m+1})^T\, \mathbf{s}^{m+1} \\
\dot{\mathbf{F}}(\mathbf{n}^m) = \begin{bmatrix} \frac{\partial f^m(n_1^m)}{\partial n_1^m} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial f^m\left(n_{S^m}^m\right)}{\partial n_{S^m}^m} \end{bmatrix} \\
\mathbf{s}^M = -2\,\dot{\mathbf{F}}(\mathbf{n}^m)\,(\mathbf{t}-\mathbf{a})
\end{gathered} \tag{2.18}$$

$\mathbf{a}^{m+1} = \mathbf{f}^{m+1}(\mathbf{W}^{m+1}\,\mathbf{a}^m + \mathbf{b}^{m+1})$, for $m$ = 0, 1, .., $M$ – 1

$\mathbf{a} = \mathbf{a}^M,\ \mathbf{a}^0 = \mathbf{p}$

where **p** is the input vector and **f** denotes the activation function. As you see the sensitivity of a layer $m$+1 is propagated backward from the last layer $M$ to the first layer.

### 2.4.2) Variations of Backpropagation Algorithms

In the previous section ‘Steepest Descent Weight Adjusting Algorithm’ is introduced. This algorithm is not fast enough for practical purpose. Some faster algorithms have been proposed to speed up the weight adjusting iterations. These algorithms can be categorized into two main group as heuristic and numerical optimization methods. We briefly explain each method in the sequel.

### 2.4.3) Heuristic Approaches

Since steepest descent weight adjusting algorithm is too slow, some heuristic algorithms were developed to speed up the learning process. Among these algorithms, use of momentum, rescaling and variable learning rate can be listed [Hagan95 p.12-2]. Though the learning can be processed faster, algorithms in this category have some drawbacks such as necessity of setting up some parameters and failure to converge; even steepest descent eventually reaches a solution [Hagan95 p.12-14]. Including a momentum term into weight adjustment is similar to applying a low-pass filter to smooth out oscillation in the output [Hagan95 p.12-9]. It helps the network to converge stably fast towards to the optimum point [Hagan95 p.12-11]. Equation (2.17) with the momentum $\gamma$ becomes

$$\begin{gathered}
\mathbf{\Delta W}^{m}(k) = \gamma\, \mathbf{\Delta W}^{m}(k-1) - (1-\gamma)\,\alpha\, \mathbf{s}^m \left(\mathbf{a}^{m-1}\right)^T \\
0 \le \gamma < 1
\end{gathered} \tag{2.19}$$

Increasing learning rate yields a faster convergence but an unstable adaption process due to skipping the optimum point. Instead of using a constant variable rate, a variable learning rate can help to flow faster in such a way that it is increased at flat regions of the performance function surface while decreasing in valleys [Hagan95 p.12-12]. In addition to the learning rate (lr) and momentum constants, learning rate increase ratio (lr_inc) and learning rate decrease ratio (lr_dec) are used to update the learning rate; if the performance function increases above a ratio then the lr is multiplied with the lr_dec, if it decreases, then lr is multiplied by lr_inc.

### 2.4.4) Numerical Optimization Approaches

Among the numerical optimization technique, the steepest descent, Newton’s method and conjugate gradient can be listed [Hagan95 p.12-14]. The steepest descent is the slowest one compared to other two approaches; Newton’s method requires computation of Hessian matrix and its inverse. The conjugate gradient is in between of steepest descent and Newton’s methods [Hagan95 p.12-14]. As a variation of Newton’s method the Levenberg-Marquardt [Hagan95 p.12-19] minimize a function which is sum of squared of other non-linear functions; this property makes it a good candidate for Backpropagation learning. The weight adjustment equation is given by

$$\Delta\mathbf{W}_k = -[\mathbf{J}^T(\mathbf{W}_k)\,\mathbf{J}(\mathbf{W}_k) + \mu_k\mathbf{I}]^{-1}\,\mathbf{J}^T(\mathbf{W}_k)\,\mathbf{e}(\mathbf{W}_k) \tag{2.20}$$

where **J** is the Jacobian matrix, **e** is the error vector, and $\mu$ is a constant for stability. The algorithm starts with a small $\mu$ (like 0.001) and it is increased by mu_inc (like multiplying by 5) if the performance function does not produce a smaller value and then the step is reapeated; on the other hand, if the function yields a smaller value then $\mu$ is divided by mu_dec for a faster convergence.

## 2.5) Literature Review on Pulmonary CT-based CADD Systems

Studies related to CT-based CADD system can be grouped under several topics such as objective of the study, lung area extraction methods, nodule detection, features used, diagnosis, assessment, preprocessing (interpolation, in general), time complexity and computation time. Objective of researches are mainly about nodule detection, diagnosis, or both detection and diagnosis, growth-rate estimation, effect of some image features on detection and diagnosis, performance of detection and diagnosis methods, reducing false detection. Receiver operating characteristic (ROC), sensitivity, specificity, and free-response receiver operating characteristic are used for assessment. Most of the studies did not mention their computation time, computation and space complexity of their algorithms. Computation time in [Fieb99] is 5 min using an O2 workstation of 250 MHz for detection of 23 nodules. In [Lee001], genetic algorithm (GA) template matching is used for detection. It takes 23 min. The same authors used a template matching algorithm alone; they reported that it takes 164 min. detection by ANN, which is a unique study — to the best of our knowledge — is employed in [Suzuki03]; the ANN is used to eliminate false candidates. Its training takes 29.9 hours with Pentium 1.7 GHz PC-based workstation whereas time to decide class of a candidate nodule is negligibly small. In the following section, we will introduce a review of detection and feature extraction methods.

### 2.5.1) Nodule Detection Methods

Nodule detection methods using pulmonary CT images can be categorized as gray-level threshold, clustering, spatial filtering, template matching, deformation methods, morphological analysis, model-based techniques [Arma04].

In [Arma99], multiple gray level (36 gray-level) thresholds are applied for detection of candidate objects. These levels are obtained with an increment of 5 in the range [5, 225]. At every threshold level, lung volumes are constructed using 10-point connectivity, which is found more suitable for anisotropic voxels. If a lung volume satisfies the threshold value of corresponding equivalent volume ($V < 4/3 \pi r^3$), then it is considered as a candidate nodule. In [Arma00], multiple-gray level threshold is utilized to obtain candidate nodules. An object less than 3 cm-diameter sphere is recognized as a nodule candidate. A rule-based technique followed by two linear discriminant analysis (LDA) classifiers is employed to decide as nodule or non-nodule using 2D and 3D features. Pixels are normalized such that -1000 HU corresponds to zero and 1000 HU corresponds to one. In [Arma00], the structures whose volume is less than a threshold are selected as candidate nodules after employing 18-point pixel connectivity to combine pixels of contiguous structures. False positive candidates are eliminated. Finally, an LDA classifier is used to identify remaining candidates as nodule or not. Number of candidate nodules is decreased using a rule-based scheme based on two features maximum eccentricity and pixel value standard deviation. The threshold used in the rules is determined empirically. Lung area extraction is performed by thorax segmentation based on multiple gray-level thresholds and then separating airways using region-growing technique. The anterior junction line is corrected, juxta-pleural or hilar structures are corrected using rolling-ball algorithm, and pixels belonging to the diaphragm are eliminated. Area under the ROC curve (Az) was reported as 93%. In [Arma03], multiple-gray-level-thresholding (region-melting) is used to identify initial lung nodule candidates based on a maximum use of maximum volume requirement. To eliminate obvious non-nodule structures a rule based technique is employed. Az was reported as 79%.

In [Gige94], a tree based on four threshold levels is constructed. The top node corresponds to the 50% of area under the histogram, and 25%, 12.5%, 6.25% in the following nodes, respectively. Geometric descriptors are perimeter, area, compactness, elongation, circularity, distance measure, and total score = area x circularity x distance were used as features. A rule-based system is used to distinguish vessels from nodules. Also in case of ambiguity, a candidate is examined using adjacent sections to determine the class of the nodule. A Gray-level thresholding technique is used to separate body from the background. A line from the center of the image to an edge of the image is guide to decide on the threshold. The lung boundary is obtained by means of gray-level histogram. The threshold at which a maximum separation between two classes is obtained, class 1 is gray value of lung area and class 2 is gray value of mediastinum and chest wall. The boundary of the lung is obtained using delineation of globally connected regions using 8-point connectivity after generating a binary image. Sensitivity was reported as 94% with 1.25 FP/case. In [Zhang05], morphologic closing, thresholding and labeling are performed followed by

LDA classifier to detect juxtapleural nodules. Detection of non-pleural nodules is achieved by an optimal thresholding (given in [Hu2001]), local shape index, and DTCNN trained by GA. Sensitivity was reported as 85.6% with 9.5 FP/case (4% per image). In [Ge2005], gradient field computation is performed to discriminate symmetric objects from non-symmetric ones and LDA is used to reduce number of false positives; sensitivity was reported as 80% with 0.37 FP/section.

In [Lou99], applying a gray level threshold followed by detection of circular shape boundaries using Hough transformation. After obtaining circular boundaries, they are combined to construct a 3D structure. A surface deformable model using location (x, y, and z coordinates), orientation (angels with the three axes), size (radius in 3 directions), squareness (in xy-plane and xz-plane), and tapering in x- and y-axis. A small tube object is used as seed and it was allowed to grow within the target object until a cost criteria is satisfied; sensitivity and specificity were reported as 88.9% and 11.1%, respectively. In [Lee001], a GA template-matching algorithm was developed. Nodules are assumed spherical; hence, a 3D Gaussian template is used as a template. The Body image is extracted by applying a threshold. GA search is performed over the 7x7 partitioned regions of the body area. Cross-correlation is used to measure the similarity between the template and the candidate structure. The lung-wall nodule detection is done by first obtaining a broad lung contour using thresholding and labeling. A semi-circular template was traversed along the lung border pixel by pixel and a similarity is computed to decide if the search region has a candidate nodule. Sensitivity was 72% with 1.1 FP/sectional image. In [Takiza03], nodules are detected using 3D Markov random field models analyzing VOI through a single plane. In [Paik2004], a statistical shape model is employed. Voxels of high density structures such as bones are set to the level of water. Isotropic voxels are obtained. All air voxels including air outside the body are thresholded and the former air voxels are negatively masked leaving only lung air voxels. Finally, a binary image is obtained. 3D orientation of image surface normal are obtained using modified Canny edge detector. Total number of surface normal vectors for a structure is calculated to measure the sphericity. Sensitivity was reported as 90% with 5.6 FP/data set.

In [Kanaza96] and [Satoh99], candidate regions are segmented by fuzzy-clustering method and nodules detected by a rule-based approach. In [Kawa99], segmentation using 3D geometric approach which is “based on deforming 3D surfaces represented level-sets” over the ROI selected interactively is used; the Az value reported as 92%. In [Kawa00], a 3D deformable surface model approach is used to extract nodules. Nodules are represented by their curvature indexes namely shape index and curvedness together with the discrete shape spectrum, which is a histogram of shape index. To further characterization of a nodule, its topological features are explored using Euler number that is composed of the number of the connected components, cavities, and holes of the 3D digital object. The type of the connectivity used is 26-connectivity. In [Fieb99], a 3D region growing is employed. A voxel having a CT value higher than a specified threshold is used to select a seed point. Process is performed twice with different seed points. A rule-based approach is used; e.g., if the extent in one dimension is more than twice the extent in another dimension, the region is assumed as a false positive. Sensitivity was reported as 100% for size ≥ 5 mm (n=10) and 95.6% for size ≥ 3 mm (n=23) with 15 FP/study (< 1 per image). The Az value reported as 97%. In [Gurc02], weighted k-means clustering with two-output class, one is for the nodule candidate and the other is for the background of the lung is utilized; different segmentation parameters are used for each segment and segmentation is done manually; sensitivity was reported as 84% with 2.53 FP/slice. In [Li2003], nodules are segmented from the background using a region growing technique and dynamic programming. In [Way2006], k-means clustering followed by 3D active contour method is applied. 26-conectivity is used in construction of candidate volumes; Az reported as 83% ±4%. In [Kuhn06], a nodule is extracted using region growing approach starting from a manually selected seed point for a manually obtained VOI followed by morphological operations.

In [Wei2006], an intelligent seed point generation method is used to speed up detection given in [Wei2002]. Nodules are detected using a volume projection method mimicking a radiologist's reading of slices in alternating forward and backward directions. Projection directions are selected using Eigen-value analysis. A Gaussian curve fitting is applied on the projected data to do a quantitative shape analysis. It is assumed that nodules are in ellipsoid shape and nodule structures have consistent projections with respect to non-nodules. A Gaussian smoothing process is applied to remove irregularities. Sensitivity reported as 67.6%, 71.9%, 6.2 FP/slice. In [Okumur98], VNQ-filter is developed to detect a nodule candidate. It is assumed that pathological shadows are given.

In [Arim04] and [Suzuki03], an ANN based detection technique (Massive Training Artificial Neural Network (MTANN)) was introduced to eliminate false nodule candidates as a final stage of elimination procedure; this technique is the only work that a special ANN model is developed for pulmonary nodule detection from CT images, to the best of our knowledge; [Arim04] reported sensitivity as 83% with 5.8 FP/scan and 84% with 5.9 FP/scan and [Suzuki03] reported sensitivity as 80.3 and 4.8 FP/patient (0.18 FP/section). MTANN works with a predefined window traversed throughout the image to obtain sub-regions. Each window is obtained shifting its center by one pixel. Training of the ANN is achieved by a modified back-propagation algorithm. The teacher image has a Gaussian-like distribution. Inputs are the pixel of the sub-regions, and the target is the central pixel of the teacher image. Each output image is scored by summation of its Gaussian weighted values. If the score exceeds a threshold then the object is considered as a nodule. STD of the Gaussian, $\sigma$, and the threshold is computed using a training set. Sub-region window size is selected as 9x9 pixels, number of hidden unit is 25, $\sigma$ is 5 pixels, training image is 19x19 pixels.

### 2.5.2) Lung Region Extraction Methods

Methods used in extraction of a lung region can be classified as threshold-based region growing, mathematical morphology, k-means clustering, and deformable model segmentation. It is reported that the thresholding approach gives better results compared to the deformable model approaches in extraction of lung area [Kim003]; the later has problem with convergence, initialization, and speed.

In [Arim04], segmentation by LDA is applied to whole body CT histogram to extract left or right lung regions. A threshold CT level is computed automatically by LDA. In [Kanaza96], the original image is transformed into the binary image using simple thresholding. In [Arma00], thorax is segmented based on multiple gray-level thresholds followed by separating airways using region-growing technique. In [Gige94], a gray-level thresholding technique is used to separate body from the background. A line from the center of the image to an edge of the image is used as a guide to decide on the threshold. The lung boundary is obtained by means of gray-level histogram. In [Wei2002], the lung volume is obtained using intensity thresholding and morphological operations. 2D region growing is applied followed by 3D region growing in [Fieb99]. K-means clustering is applied in [Gurc01] to partition pixels as lung and background; the segmentation is processed slice-by-slice.

### 2.5.3) Features

Image features are used to describe imjects based on the content of the image so that recognition can be achieved using less data and consequently smaller size search space. Features utilized in detection can be classified as photometric or geometric features. In general, photometric features are used to measure homogeneity of ROI and geometric features are used to measure circularity or sphericity. It is reported in [McNitt99] that texture features produced the best performance compared to the size and shape features.

Photometric features can be simple or complex such as gray-level (also named as attenuation or CT number), average gray value, contrast, pixel gradient, distribution of attenuation, directional variance of pixel gradient, directional cross-correlation of pixel gradient, entropy of gray-value distribution, first and second moments of gray-values, inverse difference moment of gray-values, mean of gray-values, gray-value moment ratio, standard deviation of gray-values, kurtosis of the gray value, kurtosis of the gray value histogram, max mean CT value, skewness of the gray value histogram, texture.

Geometric features can also be simple and complex such as area, average area of the candidate, distance measure, sphericity, circularity, eccentricity of the fitted ellipse, effective diameter, aspect ratio, ratio of major and minor axes, compactness, elongation, irregularity, major axes, minor axes, mean of the curvedness on each surface type, perimeter, position, volume, entropy, first and second moments , inverse difference moment, mean, moment ratio, standard deviation, shape, total score (= area x circularity x distance).

In nodule detection from CT images several features are used together. These feature combinations include 1) 13 texture features reported in [McWy99], 2) size, circularity, average CT value, variance of values, thickness, the ratio of standard area as the “proportion to the distance from lung wall to the area of ROI” [Satoh99], 3) size, shape, attenuation, distribution of attenuation, and texture [McNitt99], 4) mean, STD, area, circularity, irregularity, contrast, max mean CT value, directional variance of pixel gradient, and directional cross-correlation of pixel gradient; inverse difference moment, entropy, area, and contrast as juxtapleural features [Lee001], 5) volume, surface area, average gray value, standard deviation, skewness and kurtosis of the gray value histogram, and also area, perimeter, circularity, compactness, major and minor axes and their ratio, and eccentricity of the fitted ellipse in each cross section [Gurc02], 6) geometric descriptors are perimeter, area, compactness, elongation, circularity, distance measure, and total score = area x circularity x distance [Gige94], 7) volume, area, aspect ratio, sphericity, first and second moments, moment ratio as geometric features [Zhang05], 8) effective diameter, degree of circularity, and contrast [Li2003], 9) area, thickness, circularity, gray value, variance, position [Kanaza96], 10) eccentricity, gray-level standard deviation, average area of the candidate [Arma03], 11) 3D features used are volume, surface area, average gray-value, STD, skewness, and kurtosis of the gray value; 2D features used are maximum values of the area, perimeter, circularity, compactness, major and minor axes and their ratio, eccentricity of the ellipse fitted to the cross-section [Gurc01], 12) ratio of each surface type’s area to total surface, division number of each surface type, mean of the curvedness on each surface type [Kawa97], 13) curvature indexes, histogram of shape index, Euler number [Kawa00].

# CHAPTER 3. CONTENT BASED IMAGE REPRESENTATION[1]

Image is defined as “a two-dimensional representation of a scene” [IEEE90]. A digital image is defined as “an image that has been converted into an array of pixels, each of which has an associated value called its gray level” [IEEE90]. Sometimes the term ‘image’ is used in the meaning of ‘digital image’. An object in an image, which we name as ‘imject’ (*im*age ob*ject*), is an abstraction of a real object. A simple imject (boundary-based definition) is defined as spatially maximal region enclosed by a boundary having no common segment with another imject; that is spatially maximal disjoint region enclosed by a boundary. Also we can define and imject a mapping from the real world object to scalar space over a grid; on the other hand, imjects can be defined over a feature space. Consequently, the goal of the image representation based on the content is to find some encoding schemes that create some characteristic features to be used by the machine.

As we discussed above, an image is an abstraction itself and it needs further abstraction to be used in Computer Vision applications; in other words, images need to be ‘represented’ by their ‘contents’ using some abstraction methods. Representation of an image is achieved by encoding the image content with some distinct structures obtained from its content. If encoding is for an imject conveyed by the image, it will be named as ‘imject representation’ and this representation provides local information about the image. Encoding boundary of an imject is called ‘shape representation’, which provides some global information about the imject. A content-based representation method should be able to provide properties including uniqueness, robustness, accuracy, and invariance. Representation can be local or global. The image representation can be performed by some features like geometric (or morphological), photometric, structural or mixture of these. Morphological features give information about the structure of an object [Demi04]. They are employed to measure, for example, shape of imjects. Photometric features give information about spatial association of pixels. They are employed to measure for example homogeneity of the pixel association. As a photometric feature, texture is a connected set of pixels that occurs repeatedly in an image [Demi04 p. 7].

There is a strong relation between content representation and pattern recognition. The higher the discriminative capability of a representation posseses, the higher the performance of recognition is achieved. There are many challenges related to finding some good discriminative features. One of the main challenges in recognition of imjects is due to overlapping of their irradiance ranges. If two imjects had a totally non-overlapping range, then it would be trivial to identify them just using their range information. Another main challenge is the ‘composition of radiances’ may have some overlapping sets.

[1] Partially reprinted by permission of *IEEE International Conference on Image Processing,* pp.1493 – 1496, 2006, © [2006] IEEE; partially reprinted by permission of Proc. Of the Int’l Conf. in Image Processing, Computer Vision, & Pattern Recognition, pp.407-413, 2007, © [2007] IPCV’07; partially reprinted by permission of Proc. Of the Int’l Conf. in Image Processing, Computer Vision, & Pattern Recognition, pp.23-27, 2008, © [2008] IPCV’08.

Hence, range and possible compositions coming from this range should be studied before developing an algorithm for a recognition problem; possible set of compositions can be studied under the science of 'representation' or 'coding'.

The imject recognition problem can be defined as exploring some patterns which are composition of some primitive elements such as tonal elements [Haralic79] or composition of some features obtained from these elements such as run-length features. In other words, a pattern can be searched over the original (radiance) space or over the 'transformation space'. Primitive elements and their possible compositions can be from 1) a scalar space or 2) a spatial space. If a scalar space is employed to represent an imject, i.e. to obtain some primitive elements, some filtering functions can be used; this approach can work if imjects have some patterns which are 'range-separable'; two imjects are said to be 'range-separable' if some set of thresholds can discriminate one imject from another. The second approach which conducts pattern search over a spatial neighbor is more discriminative than the first one. A neighbor can be defined in several ways; e.g. pixels within a specific radius, pixels on a specific radius, group of pixels separated by a specific distance, and so on. A pattern which describes an imject can be explored within 1) a 'pixel neighbor', e.g. pixels on a line at a specific orientation, or 2) a 'window neighbor', e.g. 3 by 3 window, using a scalar or spatial relation. A scalar relation only considers 'magnitude-dependent' measures ignoring spatial relation such as variation of magnitudes, some statistical measures like range, mean, standard deviation, moments, entropy, etc. A spatial relation is defined based on a positional measure such as gradient, directional co-occurrence, directional order, etc.

In the following sections, the content based representation will be elaborated in more detail; a novel shape representation ('canonical polygonal representation') and a novel spectral shape feature obtained from this representation will be introduced.

## 3.1) Imject Representation

The goal of imject representation is to define some descriptive features so that the imject can be identified. The imject representation can also assist to increase computation efficiency by means of salient features. Imjects can be represented by some geometric, photometric, and topological features. In this dissertation, we will focus on first two types of features. The geometric features can be broadly classified as basic geometric features, shape features, size related features, positional features. The photometric features can be broadly classified as basic photometric features and derived photometric features.

The basic geometric features include points like corners, lines like edges, and curves like contours. The shape features to measure circularity or sphericity include effective diameter, aspect ratio, ratio of major and minor axes, eccentricity (of the fitted ellipse), compactness, elongation, major axes, minor axes, axis orientation, and symmetry. The shape features to measure curvedness include irregularity (smoothness), mean of the curvedness on each surface type, concavity and convexity, and curvature

[Carr98]. The size related features are perimeter, area, and volume. A positional feature is a distance from a specific imject or feature.

The basic photometric feature is a pixel value like intensity, gray-level, color, and transparency value. The derived photometric features include pixel gradient, optical density [Demi04 p. 9], contrast, entropy, and as the statistical photometric features: texture features, moment features, histogram features, descriptive features like mean, standard deviation, max, min, and dominant pixel value. The histogram (gray/color level distribution) features include average gray-value, kurtosis of the histogram, skewness. The moment features include first and second moments, inverse difference moment, moment ratio, center of gravity, elongation of the region, and orientation of region's axis. The texture features include coarseness, directionality, moment features, features obtained from co-occurrence matrix, Gabor features, Fourier coefficients, special texture energy, autocorrelation function, structural elements, Markov random fields model, fractal dimension, and multi-channel features.

A representation can be global or local. A global feature describes overall characteristics while a local feature describes finer details. The global features may not be a distinctive descriptor but easy to compute whereas the local features are more descriptive but more time consuming as well. Some examples of the global features are histogram features, which is obtained from whole imject, volume, area, and perimeter. Some examples of the local features that are derived from a boundary are Fourier coefficients, inflection points, edges, ridges, and curvature index.

## 3.2) Shape Representation

Shape is defined as outline of an imject. The goal of shape representation is to obtain some descriptive features so that outline of an imject can be used in identification. Therefore describing a shape by some features is a mapping from raw data of the shape to $\mathbb{R}^n$ where *n* is the dimension of a feature vector [Wolter03]. The shape features can be classified as boundary based, region based, space domain features, transform domain features [Zhang03].

The boundary based shape descriptors are classified as conventional, boundary signatures, spectral descriptors, and structural descriptors. Examples of the conventional descriptors are perimeter, area, and eccentricity. A signature primitive can be distance based such as arc length, radial length, and edge length or angel based such as local tangential angle, between-radial angle, between-edge angle, and centroid-edge angle. Also it can be distance-angle based such as radial distance function, curvature function [Kawa00], and turning function. The spectral descriptors include Fourier coefficients, wavelet coefficients such as Gabor wavelet coefficients. The structural descriptors can be polygonal descriptors, skeleton, shock graph, topological features such as Delaunay triangulation [Demi04 p. 8]. The domain based shape descriptors include parametric contours such as chain codes, curve approximations such as splines, stochastic methods such as Hidden Markov Model and autoregressive model. The region based descriptors are area, compactness, Euler number, geometric moments, Zernike moments, pseudo-Zernike moments, Legendre moments, and grid method.

## 3.3) Canonical Polygonal Representation

Numerous models have been defined for shape representation. Each representation has its own primitives employed to find the best match. Shape primitives can be grouped based on utilizing radial distance from an origin, the edge segment lengths, and the angle between connected segments [Chang91, Chen95, Huang96, Wang03, Wu2001, Chuang96, Maes91, Nishida98]. Representing shapes can be achieved using vertex-based or edge-based methods. According to [Chen95], The former has less computation time while the latter is more accurate. Among the edge-based methods, polygonal representation of shapes has been studied extensively [Chen95]. Matching of polygons has been achieved by string representation using the three primitives mentioned above [Wu2001, Maes91]. These three primitives can be organized in different combinations to obtain a unique representation. An orientation invariant representation has been obtained by shifting [Maes91, Huang96, Arkin91] or using max/min distance [Nishida98, Bernier03].

We proposed a new polygonal shape representation which is called ‘canonical polygonal representation’ (CPR), an efficient algorithm to compute a unique representation of a polygon under rigid transformation and can be applied to affine transformation with some additional processes [Soysal06]. The computational complexity for matching of two polygons with $n$ vertices has been reported as either O($n^2$) [Schreiber90] or O($n^2$ log $n$) [Arkin91, Maes90] to the best of our knowledge. Our proposed method has O($n$ log $n$) computational complexity for representation and Θ($n$) for matching. Hence, if shapes are stored with the proposed representation, matching can be done in Θ($n$), which is the optimum, and online matching can be achieved in O($n$ log $n$) after obtaining vertices.

Our novel shape representation exploits two primitives which are radial distance and its corresponding edge distance for a given polygon $P$. For each vertex of the polygon, a distance vector $\delta_i$ is obtained using these two distances and then all vectors are cyclically ordered in a specified direction to get a canonical string representation $P_C$ of the polygon. To handle the starting point problem, a normalized string $P_N$ is obtained from the $P_C$ using *Normalize String* algorithm proposed below.

### 3.3.1) Canonical Representation

Let $P = \langle (x_1, y_1), (x_2, y_2), .. (x_n, y_n) \rangle$ denotes a polygon of $n$ vertices ordered in a specified direction and $c = (x_c, y_c)$ denotes the centroid of $P$ given by (3.1). Let $\delta_i = (\delta_{1i}, \delta_{2i})$ be a distance vector where the edge $\delta_{1i}$ (from a vertex $(x_i, y_i)$ to a vertex $(x_{i+1}, y_{i+1})$) and the radial $\delta_{2i}$ (from $(x_i, y_i)$ to $(x_c, y_c)$) distances defined in equations (3.2) and (3.3) below.

$$c = \left( \frac{\sum_1^n x_i}{n}, \frac{\sum_1^n y_i}{n} \right) \tag{3.1}$$

$$\delta_{1i} = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \qquad (3.2)$$

$$\delta_{2i} = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \qquad (3.3)$$

We define the ‘canonical polygonal representation’ (CPR) of *P* as

$$P_{\mathrm{C}} = \langle \delta_1, \delta_2, .., \delta_n \rangle \qquad (3.4)$$

where $\langle ... \rangle$ defines a sequence and the ‘normalized canonical polygonal representation’ (NCPR) of *P* as

$$P_{\mathrm{N}} = \langle \delta_a, \delta_{a \oplus 1}, .., \quad \delta_{a \oplus (n-1)} \rangle \qquad (3.5)$$

which is obtained from $P_{\mathrm{C}}$ as explained below, where the anchor index *a* defines the starting vertex which is unique for the polygon; note that $1 \leq i \oplus n = \mathrm{mod}(i + n, n) \leq n$ for $1 \leq i \leq n$. The Figure 3.1 shows components of a distance vector for a vertex.

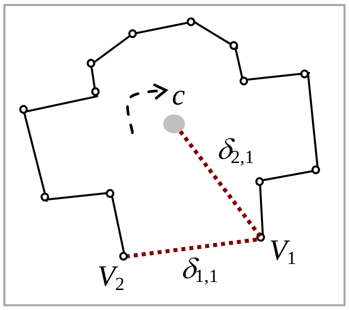


Figure 3.1 The components of a distance vector ($V_i$ is a vertex of the polygon)

### 3.3.2) Normalized Canonical Representation

We proposed an algorithm *Normalize String* to compute NCPR $P_{\mathrm{N}}$ of *P* [Soysal06]. The algorithm searches iteratively for a unique starting vertex by comparing distance vectors. The algorithm begins by finding all maximal distance vectors; each such vector is called a ‘head’ *H*. If more than one max distance vectors are found at the first iteration, then the string is partitioned into sub-strings, called ‘segment’ *S*. Each segment is a sequence starting from a head to the next head; the sequence excluding the head is called ‘tail’ *T* of the segment. Therefore, each segment is composed of a head and a tail. After construction of segments, the algorithm searches for the max segment by comparing their tails. If more than one max segment is found, then max segments are merged with smaller ones making a longer segment. This comparison-and-merging process continues until a unique max segment is found. The formal definitions are given in the sequel.

Comparison of distance vectors: Given two distance vectors $\delta_1$ and $\delta_2$ and a threshold $\tau > 0$, the inequality $\delta_1 > \delta_2$ is true if one of these two conditions are satisfied:

$$\begin{aligned} &1)\quad \delta_{11} - \delta_{12} > \tau \\ &2)\quad |\delta_{21} - \delta_{22}| \le \tau \end{aligned} \tag{3.6}$$

Comparison of sequences: Given two sequences $T_1 = \langle \delta_1^1,\ \delta_2^1, .., \delta_r^1 \rangle$ and $T_2 = \langle \delta_1^2,\ \delta_2^2, .., \delta_r^2 \rangle$, the inequality $T_1 > T_2$ is true if one of these conditions are satisfied:

$$\begin{aligned} &1)\quad \delta_1^1 > \delta_1^2 \\ &2)\quad \delta_K^1 = \delta_K^2 \text{ and } \delta_{j+1}^1 > \delta_{j+1}^2, \text{for } 1 \le K \le j \text{ such that } 1 \le j < r \end{aligned} \tag{3.7}$$

Computation of max segment: Let $H = H_i = \{\delta_i : \delta_i = \max_{j=1,2,..,n} \delta_j\}$ for some $n > 0$, denotes a 'head' and $T$ denotes a 'tail', which is the sub-sequence between two successive heads $H_t$ and $H_{t\oplus 1} \in H$ for $t$ = 1, 2, .., $|H|$, and let $\hat{S}_t = \langle H_t, T_t \rangle$ denotes a segment of $P_\mathrm{C}$. We compute max of two segments $\hat{S}_1 = \langle H_1, T_1 \rangle$ and $\hat{S}_2 = \langle H_2, T_2 \rangle$ using

$$\operatorname{argmax}(\hat{S}_1, \hat{S}_2) = \begin{cases} \operatorname{argmax}(|T_1|, |T_2|), & |T_1| \ne |T_2| \\ \quad \operatorname{argmax}(T_1, T_2), & |T_1| = |T_2| \end{cases} \tag{3.8}$$

Computation of parent segments: Let $\hat{S}^{(k)}$ and $m_k$ denote the set of segments and number of segments at iteration $k$, respectively, and $I^{(k)}$ denotes the head index of each segment to be computed at iteration $k$ + 1. Then a parent segment is computed by

$$\begin{aligned} \hat{S}^{(k)} = f\left(\hat{S}^{(0)}, I^{(k-1)}\right) &= \langle \hat{S}_1^{(k)}, \hat{S}_2^{(k)}, .., \hat{S}_{m_{k-1}}^{(k)} \rangle \\ &= \langle \langle H_1^{(k)}, T_1^{(k)} \rangle, \langle H_2^{(k)}, T_2^{(k)} \rangle, .., \langle H_{m_{k-1}}^{(k)}, T_{m_{k-1}}^{(k)} \rangle \rangle \end{aligned} \tag{3.9}$$

$$\hat{S}^{(0)} = \langle \langle \delta_1 \rangle, \langle \delta_2 \rangle, .., \langle \delta_n \rangle \rangle \tag{3.10}$$

$$H_t^{(k)} = \delta_{I^{(k-1)}(t)} \tag{3.11}$$

$$T_t^{(k)} = \delta_{I^{(k-1)}(t)\oplus 1}, \delta_{I^{(k-1)}(t)\oplus 2}, .., \delta_{I^{(k-1)}(t\,\oplus\,1)-1} \tag{3.12}$$

for $t = 1,2, .., m_{k-1}$

$$I^{(k)} = \left\{ I^{(k-1)}(t') : t' = \operatorname*{argmax}_{1 \le j \le m_{k-1}} \left(\hat{S}_j^{(k)}\right) \right\} \tag{3.13}$$

where $m_k = \left| I^{(k)} \right|$.

### 3.3.2.1) Algorithm Normalize String

Strategy: Divide and conquer cyclically while constructing a tree in bottom-up form (An operation such as find, merge, and compare are done cyclically.).

Goal: To find a unique ordering among the cyclic permutations of $P_C$.

Input: A cyclic string $P_C = \langle \delta_1, \delta_2, .., \delta_n \rangle$.

Output: The normalized form $P_N$ of string $P_C$.

1. [Initialization]: Set of initial segments:
   $\hat{S}^{(0)} = \langle \hat{S}_1^{(0)}, \hat{S}_2^{(0)}, .., \hat{S}_n^{(0)} \rangle = \langle \langle \delta_1 \rangle, \langle \delta_2 \rangle, .., \langle \delta_n \rangle \rangle$.
2. [Terminal layer construction]: Find all heads of $\hat{S}^{(0)}$ and store their indices in $I^{(0)} = \left\{ argmax_{1 \le j \le m_{k-1}} \left( \hat{S}_j^{(0)} \right) \right\}$ and compute $m_0 = \left| I^{(0)} \right|$. Make $I^{(0)}$ the terminal layer of the tree.
3. [Parent layer construction: $I^{(k)}, 1 \le \mathrm{k} \le \mathrm{K}$]: Repeat until $m_{k-1} = 1$ or a symmetrical case $(m_{k-1} = m_k)$ will be decided.
   a. Compute the set $\hat{S}^{(k)}$ of parent segments from the set $\hat{S}^{(k-1)}$ of child segments by equation (3.9).
   b. Find the max segments $\hat{S}_j^{(k)} \in \hat{S}^{(k)}$, and store their index in $I^{(k)}$ by equation (3.13) using the following operations:
      i. First compare the segments in length,
      ii. If more than one max length segment exist, then compare these segments pair-wise using equation (3.8).
4. $a = I^{(K)}(1)$, $P_{\mathrm{N}} = \langle \delta_a, \delta_{a \oplus 1}, .., \ \delta_{a \oplus (n-1)} \rangle$.

As an example, let a cyclic string

$$P_{\mathrm{C}} = \langle (3, 1), (2, 1), (5, 1), (3,1), (4, 1), (5, 1) \rangle$$

then $P_{\mathrm{N}}$ will be computed as

$$P_{\mathrm{N}} = \langle (5, 1), (3, 1), (4, 1), (5, 1), (3, 1), (2, 1) \rangle$$

where segments are $\hat{S}^{(1)} = \langle (5, 1), (3, 1), (4, 1) \rangle$ and $\hat{S}^{(2)} = \langle (5, 1), (3, 1), (2, 1) \rangle$ with $H = \langle (5, 1) \rangle$ and $T_1 = \langle (3, 1), (4, 1) \rangle$ and $T_2 = \langle (3, 1), (2, 1) \rangle$, and $\delta_a = \delta_3$.

A symmetrical example is $P_{\mathrm{C}} = \langle (5, 1), (3, 1), (4, 1), (5, 1), (3, 1), (4, 1) \rangle$, so $\delta_a$ will be any of head $\delta_3$ or $\delta_6$. For an illustration of the algorithm see the Figure 3.2.

**Theorem** (Complexity of the Algorithm): Let $k$ = 0, 1, 2, .., $K$ ($\ge$ 0) denotes the iteration count, $T_{\mathrm{L}}$ denotes computation complexity of comparison of some tails in length, $T_{\mathrm{A}}$ denotes computation complexity of pair-wise comparison of the tails, $T_{\mathrm{I}}$ denotes computation complexity of updating $I^{(k)}$, and $T$ denotes the total computation complexity. The complexity of the algorithm normalize string is $O(n \ \log_2 n)$.

Proof: The number of maximum segments will be halved at successive iteration for a worst-case scenario. Hence

$$m_k = \frac{m_{k-1}}{2} \tag{3.14}$$

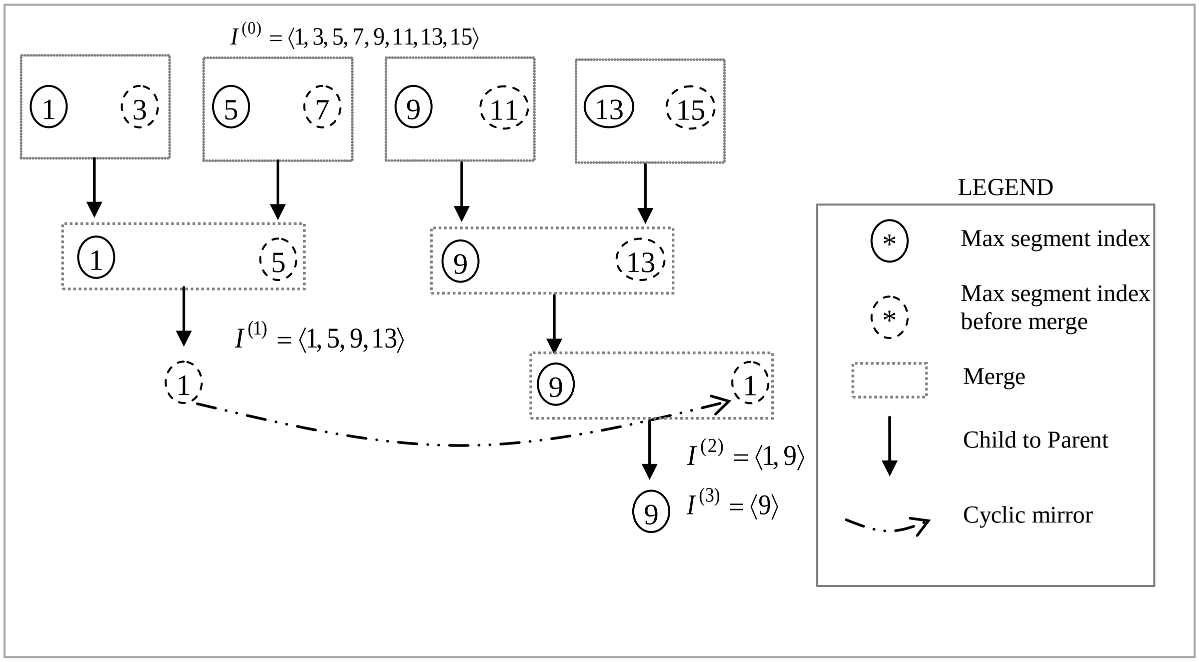

$I^{(0)} = \langle 1,3,5,7,9,11,13,15 \rangle$
1
3
5
7
9
11
13
15
1
5
9
13
$I^{(1)} = \langle 1,5,9,13 \rangle$
1
9
1
$I^{(2)} = \langle 1,9 \rangle$
9
$I^{(3)} = \langle 9 \rangle$
LEGEND
*
Max segment index
*
Max segment index before merge
Merge
Child to Parent
Cyclic mirror

Figure 3.2 Illustration of the algorithm Normalize String

Using equation (3.14) and $m_0 = \left|I^{(0)}\right|$ which is obtained in computation of max of $\hat{S}^{(0)} = \langle\langle\delta_1\rangle, \langle\delta_2\rangle, \ldots, \langle\delta_n\rangle\rangle$, we obtain

$$m_K = \frac{m_{K-1}}{2} = \frac{m_0}{2^{\mathrm{K}}} = 1 \Longrightarrow K = \log_2 m_0 \tag{3.15}$$

If we iterate $k$, we obtain equation (3.16) as given below

$k = 0$: $T_{\mathrm{L}} = 0$, $T_{\mathrm{A}} = n$, $T_{\mathrm{I}} = m_0$, hence $T = n + m_0$

$k = 1$: $m_1 = m_0/2^1$, $T_{\mathrm{L}} = m_0$, $T_{\mathrm{A}} = n$, $T_{\mathrm{I}} = m_1$, hence $T = m_0 + n + m_0/2^1$

$k = 2$: $m_2 = m_1/2^1 = m_0/2^2$, $T_{\mathrm{L}} = m_1$, $T_{\mathrm{A}} = n/2$, $T_{\mathrm{I}} = m_2$, hence $T = m_0/2^1 + n/2 + m_0/2^2$

..

$k = k'$: $m_{k'} = m_{k'-1}/2^1 = m_0/2^{k'}$, $T_{\mathrm{L}} = m_{k'-1}$, $T_{\mathrm{A}} = n/2$, $T_{\mathrm{I}} = m_{k'}$, hence $T = m_{k'-1} + n/2 + m_0/2^{k'}$

..

$k = K$: $m_K = m_{K-1}/2^1 = m_0/2^K$, $T_{\mathrm{L}} = m_{K-1}$, $T_{\mathrm{A}} = n/2$, $T_{\mathrm{I}} = m_K$, hence $T = m_{K-1} + n/2 + m_0/2^K$

$$T = n + m_0 + \sum_{k=1}^{K} \left(T_L^{(k)} + T_A^{(k)} + T_I^{(k)}\right) \tag{3.16}$$

Since $T_A^{(1)} = n = \frac{n}{2} + \frac{n}{2}$, $T_A^{(k)} = \frac{n}{2}$ for $k > 1$, and $m_k = \frac{m_{k-1}}{2}$ rearranging equation (3.16) we get

$$
\begin{aligned}
T &= n + m_0 + \frac{n}{2} + \sum_{k=1}^{K} \left(m_{k-1} + \frac{n}{2} + m_k\right) \\
&= n + m_0 + \frac{n}{2} + K\frac{n}{2} + \frac{3}{2}\sum_{k=1}^{K} m_{k-1} \\
&= n + m_0 + \frac{n}{2}(K+1) + \frac{3}{2}(m_0 + m_1 + .. + m_K) \\
&= n + m_0 + \frac{n}{2}(K+1) + \frac{3}{2}\left(m_0 + \frac{m_0}{2} + .. + \frac{m_0}{2^K}\right) \\
&= n + m_0 + \frac{n}{2}(K+1) + \frac{3}{2}m_0 \sum_{k=0}^{K} \frac{1}{2^k} \\
&= n + m_0 + \frac{n}{2}(K+1) + 3m_0 - \frac{3}{2}
\end{aligned}
\tag{3.17}
$$

Substituting equation (3.15), we get

$$T = O(n \ \log_2 m_0) \tag{3.18}$$

Since $m_0 \le n$, then $T = O(n \ \log_2 n)$.

Justification of Worst-case scenario: If the number of segments is not halved, then one of the segments will grow up at successive iteration faster than others, and it will be identified as the unique max. Hence, in the worst-case scenario, at every successive iteration, segments must grow in balance until all of them will merge into one segment□.

### 3.3.3) Robustness of the Representation

In this section, we elaborate on the robustness conditions of the canonical representation in case of some vertices are located with an error $e_i = \left(e_{x,i}, e_{y,i}\right)$. Let $V_i = (x_i, y_i)$ and $\tilde{V}_i = (\tilde{x}_i, \tilde{y}_i) = V_i + e_i$ denote actual location of a vertex and its new observed location after perturbation by the error, respectively.

Let $V = (v_1, v_2, .., v_n)$ and $\tilde{V} = (\tilde{v}_1, \tilde{v}_2, .., \tilde{v}_n)$ denote the true sequence of vertices (say, in clockwise direction) and the observed clockwise vertex sequence, respectively. Under the robustness conditions that we will identify in the sequel, the order of $V$ and $\tilde{V}$ must be satisfied and moreover $\tilde{v}_i \neq \tilde{v}_j$ for $i$ = 1, 2, …, $n$ and $i \neq j$. Thus, the anchor $\tilde{\delta}_{\tilde{a}}$ index $\tilde{a}$ computed by the canonical representation algorithm from $\tilde{\delta}$ using a threshold $\tau > 0$ must be the same as the actual anchor $\delta_a$ index $a$ computed using $\tau = 0$.

Let $dis(v_i, \tilde{v}_i) = \sqrt{(x_i - \tilde{x}_i)^2 + (y_i - \tilde{y}_i)^2}$ denotes the distance between $v_i$ and $\tilde{v}_i$ for every *i* = 1, 2, …, *n*. The $\tilde{V}$ is said to be within '*ε-error* of' $V$ if the distance $dis(v_i, \tilde{v}_i)$ is at most *ε* for some *ε > 0.*

We also say that the corresponding distance vector sequence $\tilde{\delta} = \left(\tilde{\delta}_1, \tilde{\delta}_2, .., \tilde{\delta}_n\right)$ is within '*ε-error* of' $\delta = (\delta_1, \delta_2, .., \delta_n)$.

The new centroid will move to a new location $\tilde{c}_i = (\tilde{x}_c, \tilde{y}_c)$ due to the error. The centroid is given by

$$\tilde{x}_c = \frac{\sum_{i=1}^{n} \tilde{x}_i}{n} = \frac{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} e_{x,i}}{n} = x_c + \bar{e}_x \tag{3.19}$$

similarly,

$$\tilde{y}_c = y_c + \bar{e}_y \tag{3.20}$$

where

$$\bar{e}_x = \frac{\sum_{i=1}^{n} e_{x,i}}{n} \text{ and } \bar{e}_y = \frac{\sum_{i=1}^{n} e_{y,i}}{n} \tag{3.21}$$

The new distance vector $\tilde{\delta}_i = \left(\tilde{\delta}_{1i}, \tilde{\delta}_{2i}\right)$ will be computed using

$$\tilde{\delta}_{1i} = \sqrt{(\tilde{x}_{i+1} - \tilde{x}_i)^2 + (\tilde{y}_{i+1} - \tilde{y}_i)^2} \tag{3.22}$$

$$\tilde{\delta}_{2i} = \sqrt{(\tilde{x}_i - \tilde{x}_c)^2 + (\tilde{y}_i - \tilde{y}_c)^2} \tag{3.23}$$

Let an observed vertex $\tilde{V}_i$ be located at the vicinity of an actual vertex $V_i$ bounded by radius $\varepsilon$. Let an equivalence class $C(d|\varepsilon)$ of a distance value $d$ for an error $\varepsilon$ specified denotes a set of all distance values $\tilde{d}$ in the closed interval of the segment $[d - 2\varepsilon, d + 2\varepsilon]$. That is, all of the distance values $\tilde{d}$ within the interval maps to $d$. Formally, $C(d|\varepsilon) = \left\{\tilde{d} : \tilde{d} \in [d - 2\varepsilon, d + 2\varepsilon]\right\}$.

The robustness conditions of the representation using a threshold $\tau > 0$ for $i, j$ = 1, 2, …, $n$ and $\forall\, i \neq j$ are

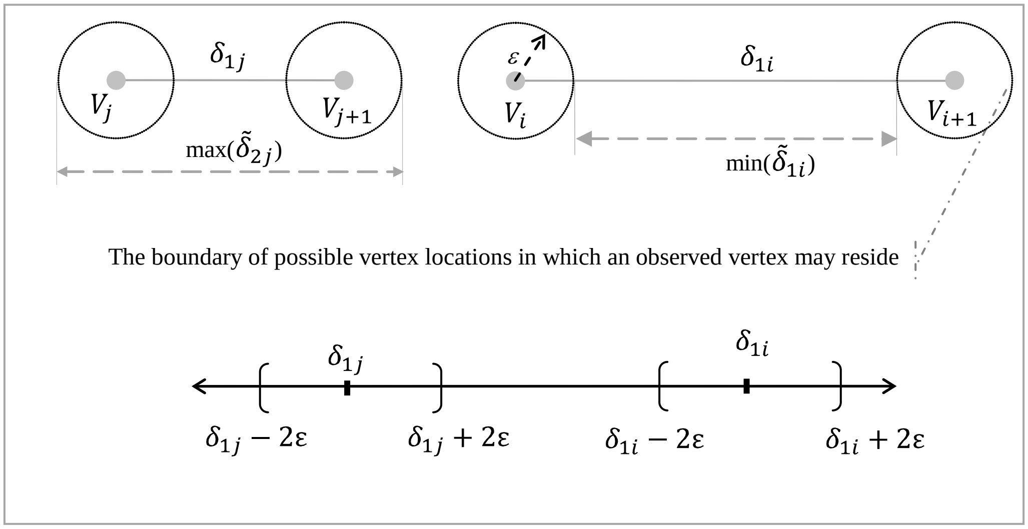
(1) Distance distinction: $\delta_{1i} \neq \delta_{1j} \Leftrightarrow \tilde{\delta}_{1i} \neq \tilde{\delta}_{1j}$ and $\left|\tilde{\delta}_{1i} - \tilde{\delta}_{1j}\right| \leq \tau \Leftrightarrow \delta_{1i} = \delta_{1j}$

(2) Order preserving: $\tilde{\delta}_{1i} - \tilde{\delta}_{1j} > \tau \Leftrightarrow \delta_{1i} > \delta_{1j}$

Figure 3.3 is used for illustration of lemmas stated below. In all the lemmas below, we assume that $\tilde{\delta} = \left(\tilde{\delta}_1, \tilde{\delta}_2, .., \tilde{\delta}_n\right)$ is within *ε-error* of $\delta$.

Lemma-1: $\tilde{\delta}_{1i\,min} = \delta_{1i} - 2\varepsilon$ and $\tilde{\delta}_{1i\,max} = \delta_{1i} + 2\varepsilon \;\forall\, i = 1, 2, .., n$.

We remark that the condition $\delta_{1i} > 2\varepsilon$ must be satisfied under the assumption 'within *ε-error*'. Recall the assumption that the observed vertex sequence $\tilde{V}$ preserves the ordering given by the actual vertex sequence $V$; that is, we will not get any observed vertex sequence such as $\tilde{V} = (\tilde{v}_1, \tilde{v}_2, .., \tilde{v}_{i-1}, \tilde{v}_{i+1}, \tilde{v}_i, .., \tilde{v}_n)$. Also, the robustness conditions must assure that the observed vertices are distinctive. Now, if we have some actual vertices $v_i$ and $v_{i+1}$ such that $\delta_{1i} = dis(v_i, v_{i+1}) \leq 2\varepsilon$, then we might have $\tilde{v}_i$ = $\tilde{v}_{i+1}$ or an out-of-order vertex sequence shown above.



$\delta_{1j}$
$V_j$
$V_{j+1}$
max($\tilde{\delta}_{2j}$)
$\varepsilon$
$V_i$
$\delta_{1i}$
$V_{i+1}$
min($\tilde{\delta}_{1i}$)
The boundary of possible vertex locations in which an observed vertex may reside
$\delta_{1j}$
$\delta_{1i}$
$\delta_{1j} - 2\varepsilon$
$\delta_{1j} + 2\varepsilon$
$\delta_{1i} - 2\varepsilon$
$\delta_{1i} + 2\varepsilon$

Figure 3.3 Illustration for lemmas

Lemma-2: Robustness conditions (1) and (2) with $\tau = 4\varepsilon$ are satisfied iff condition (3.24) is satisfied for *i*, *j* = 1, 2, .., *n*; *i* ≠ *j*

$$\delta_{1i} > \delta_{1j} \Leftrightarrow \delta_{1i} - \delta_{1j} > 8\varepsilon \qquad (3.24)$$

Proof: Let's show that (A) If the robustness conditions (1) and (2) are true, then (3.24) must be true:

The condition (2), $\tilde{\delta}_{1i} - \tilde{\delta}_{1j} > \tau = 4\varepsilon \Leftrightarrow \delta_{1i} > \delta_{1j}$, is true, that is (A.1): $\delta_{1i} > \delta_{1j} \Rightarrow \tilde{\delta}_{1i} - \tilde{\delta}_{1j} > 4\varepsilon$ is true; then, $\tilde{\delta}_{1i\,min} - \tilde{\delta}_{1j\,max} > 4\varepsilon \;\Rightarrow \delta_{1i} - 2\varepsilon - \delta_{1j} - 2\varepsilon > 4\varepsilon \;\Rightarrow \delta_{1i} - \delta_{1j} > 8\varepsilon$. Now, (A.2): if $\delta_{1i} - \delta_{1j} > 8\varepsilon$, then $\delta_{1i} - \delta_{1j} > 0 \Rightarrow \delta_{1i} > \delta_{1j}$, which is obvious.

(B) Now, let's show that if condition (3.24) is true, then the robustness conditions (1) and (2) must be true:

That is if $\delta_{1i} > \delta_{1j} \Leftrightarrow \delta_{1i} - \delta_{1j} > 8\varepsilon$ is true, then condition (2), $\tilde{\delta}_{1i} - \tilde{\delta}_{1j} > \tau \Rightarrow \delta_{1i} > \delta_{1j}$, must be true. (B.1): $\tilde{\delta}_{1i} - \tilde{\delta}_{1j} > \tau \;\Rightarrow \tilde{\delta}_{1i\,\min} - \tilde{\delta}_{1j\,\max} > 4\varepsilon \;\Rightarrow \delta_{1i} - 2\varepsilon - \delta_{1j} - 2\varepsilon > 4\varepsilon \;\Rightarrow \delta_{1i} - \delta_{1j} > 8\varepsilon \Rightarrow \delta_{1i} > \delta_{1j}$, and (B.2): $\delta_{1i} > \delta_{1j} \;\Rightarrow \tilde{\delta}_{1i} - \tilde{\delta}_{1j} > \tau$ must be true. Now, $\delta_{1i} > \delta_{1j} \;\Rightarrow \delta_{1i} - \delta_{1j} > 8\varepsilon \Rightarrow \tilde{\delta}_{1i\,\min} - 2\varepsilon - \tilde{\delta}_{1j\,\max} - 2\varepsilon > 8\varepsilon \Rightarrow \tilde{\delta}_{1i\,\min} - \tilde{\delta}_{1j\,\max} > 4\varepsilon \Rightarrow \tilde{\delta}_{1i} - \tilde{\delta}_{1j} > 4\varepsilon$.

Now, let's show the condition (1) is true when $\delta_{1i} - \delta_{1j} > 8\varepsilon$ is true; (C.1): $\delta_{1i} \neq \delta_{1j} \Rightarrow \tilde{\delta}_{1i} \neq \tilde{\delta}_{1j}$ must be true. $\delta_{1i} \neq \delta_{1j} \Rightarrow \tilde{\delta}_{1i\,\min} - \tau > \tilde{\delta}_{1j\,\max} \Rightarrow \tilde{\delta}_{1i\,\min} - \tilde{\delta}_{1j\,\max} > \tau \Rightarrow \tilde{\delta}_{1i} \neq \tilde{\delta}_{1j}$. (C.2): $\tilde{\delta}_{1i} \neq \tilde{\delta}_{1j} \Rightarrow \delta_{1i} \neq \delta_{1j}$ must be true. $\tilde{\delta}_{1i} \neq \tilde{\delta}_{1j} \Rightarrow \tilde{\delta}_{1i\,\min} + 2\varepsilon \neq \tilde{\delta}_{1j\,\max} - 2\varepsilon \Rightarrow \delta_{1i} \neq \delta_{1j}$. (C.3): $\left|\tilde{\delta}_{1i} - \tilde{\delta}_{1j}\right| \leq \tau \Rightarrow \delta_{1i} = \delta_{1j}$ must be true. $\left|\tilde{\delta}_{1i} - \tilde{\delta}_{1j}\right| \leq \tau \Rightarrow \left|\tilde{\delta}_{1i\,\min} - \tilde{\delta}_{1j\,\max}\right| \leq 4\varepsilon \Rightarrow \left|\delta_{1i} - 2\varepsilon - \delta_{1j} - 2\varepsilon\right| \leq 4\varepsilon \Rightarrow \delta_{1i} - \delta_{1j} \leq 8\varepsilon \Rightarrow \delta_{1i} \leq \delta_{1j}$, since $\delta_{1i} \geq \delta_{1j}, \Rightarrow \delta_{1i} = \delta_{1j}$. (C.4): $\delta_{1i} = \delta_{1j} \Rightarrow \left|\tilde{\delta}_{1i} - \tilde{\delta}_{1j}\right| \leq \tau$ must be true. $\delta_{1i} = \delta_{1j} \Rightarrow \tilde{\delta}_{1i\,\min} + 2\varepsilon = \tilde{\delta}_{1j\,\max} - 2\varepsilon \Rightarrow \tilde{\delta}_{1i\,\min} - \tilde{\delta}_{1j\,\max} = -4\varepsilon \Rightarrow \left|\tilde{\delta}_{1i} - \tilde{\delta}_{1j}\right| \leq 4\varepsilon$.

**Theorem-1**: The robustness of the representation is satisfied for edge distance comparison iff the conditions (1) and (2) are satisfied.

Proof: See lemma 1 and lemma2.

**Theorem-2**: The robustness of the representation is satisfied for radial distance comparison iff the conditions (1) and (2) are satisfied by replacing $\varepsilon$ by $\varepsilon'$ and $\tau$ by $\tau' = 4\varepsilon'$.

Proof: See lemma-3 and lemma-4.

Lemma-3: In existence of an error at some vertices, possible locations of the center $\tilde{c}$ of the observed vertices are bounded by a circle centered at the actual location $c$ with a radius $\varepsilon$. That is

$$\|c - \tilde{c}\| \leq \varepsilon \tag{3.29}$$

Proof: Given $\tilde{V}_i = V_i + e_i$ and $\|e_i\| \leq \varepsilon \ \forall \ i = 1, 2, \ldots, n$ and recall $\bar{e} = \tilde{c} - c$. From (3.21), $\bar{e} = \frac{\sum_{i=1}^{n} e_i}{n} = \left(\frac{\sum_{i=1}^{n} e_{x,i}}{n}, \frac{\sum_{i=1}^{n} e_{y,i}}{n}\right)$. So, $\|\bar{e}\| = \frac{\|\sum_{i=1}^{n} e_i\|}{n} \leq \frac{\sum_{i=1}^{n}\|e_i\|}{n} \leq \frac{\sum_{i=1}^{n} \varepsilon}{n} = \varepsilon$. Hence, $\|\bar{e}\| \leq \varepsilon$.

Lemma-4: For the radial distances $\left(\tilde{\delta}_{21}, \tilde{\delta}_{22}, .., \tilde{\delta}_{2n}\right)$, we have $\tilde{\delta}_{2i\,\min} = \delta_{2i} - \left(1 - \frac{1}{n}\right) 2\varepsilon$, $\tilde{\delta}_{2i\,\max} = \delta_{2i} + \left(1 - \frac{1}{n}\right) 2\varepsilon$. Let us define $\varepsilon' = \left(1 - \frac{1}{n}\right) \varepsilon$.

Proof: $\tilde{\delta}_{2i\,\min} = \delta_{2i} - \|e_i\| - \|\bar{e}\|$; that is, all the error vectors are in the same direction except $e_i$ that is in the opposite direction, formally, $e = e_1 = e_2 = .. = -e_i = e_{i+1} = .. = e_n$. Then, the total error $\bar{e} = \frac{\sum_{k=1,k\neq i}^{n} e_k}{n} - \frac{e_i}{n} = \frac{n-1}{n}\varepsilon - \frac{n}{n}\varepsilon = \frac{n-2}{n}\varepsilon$. So, $\tilde{\delta}_{2i\,\min} = \delta_{2i} - \varepsilon - \frac{n-2}{n}\varepsilon = \delta_{2i} - \left(1 - \frac{1}{n}\right) 2\varepsilon$. Similarly, $\tilde{\delta}_{2i\,\max} = \delta_{2i} + \|e_i\| + \|\bar{e}\| = \delta_{2i} + \varepsilon + \frac{n-2}{n}\varepsilon = \delta_{2i} + \left(1 - \frac{1}{n}\right) 2\varepsilon$.

**Theorem-3**: The robustness of the representation is satisfied iff Theorem-1 and Theorem-2 are satisfied.

## 3.4) Pattern Features

As we mentioned earlier, imjects can be recognized by some patterns exploring scalar or spatial relations. In some cases, employing the primitive content or higher level patterns directly may not be feasible due to the curse of dimensionality. Also, the primitive content of an image data may not provide enough information about the imject to discriminate from other imjects. Instead of using the raw data, some salient features can be extracted to reduce the computation time and the amount of data to be used in recognition. Therefore, the higher level content may need to be transformed into some abstract patterns called features. Features can increase "between-class pattern variability" while "minimizing the within-class variability" [LeonC98 p. 7]. A feature is a salient descriptor of a pattern. The feature extraction can be achieved by many methods such as obtaining histograms, using some spatial filters as in edge patterns, obtaining shape representation as in canonical representation of a polygon, or computing texture features from co-occurrence. In this dissertation, we utilized two classes of pattern features: 1) geometric features and 2) photometric features.

### 3.4.1) Geometric Features

In this dissertation, we employed three types of shape features: 1) Spectral shape feature, 2) circularity features including elongation, eccentricity, circularity by radial distance, circularity by CPR, and circularity by spectral feature, 3) regional features including normalized Euler's number, compactness, and convexity. We first introduce our novel shape feature and then rest of the features [Sonka98, Gallo75, Haralic73] is explained; among these, we proposed three new circularity measures, which are circularity by CPR, circularity by spectral feature, and circularity by radial distance.

### 3.4.1.1) Spectral Shape Feature

In the previous section, we introduced Normalized Canonical Polygonal Representation. Here, we introduce a novel spectral shape feature [Soysal08] which is obtained from NCPR $P_\mathrm{N}$ of a polygon [Soysal07]. The spectral feature is a set of Fourier coefficients of a shape function derived from $P_\mathrm{N}$. As a spectral descriptor, the Fourier coefficients that are obtained by discrete Fourier transform can be employed in shape representation. The performance of the Fourier descriptor is acknowledged in [Zhang03].

### 3.4.1.2) Derivation of the Shape Function

In this section, we describe how to derive a shape function which is used to obtain the proposed shape feature. We propose three methods to derive a descriptive function for the shape. After obtaining the function, an interpolation is necessary to obtain a normalized axis with equal intervals for the discrete Fourier transform. The shape function derivation methods proposed are:

1. Values for the dependent variable $y$ are obtained from the set of $\delta_{1i}$ and values for the independent variable $x$ are obtained from the set of $\delta_{2i}$, formally:

$$\hat{y}[x_i] = \delta_{1i} \tag{3.25}$$

where $x_i = c\left(\sum_{j=1}^{i} \delta_{2,j} - \delta_{2,1}\right)$ for $i = 1, 2, .., n$, and c > 0.

2. $y$ is a combination of the sequence $\{\delta_{1i}\}$ followed by the sequence $\{\delta_{2i}\}$, and $x$ is obtained from indexes, formally:

$$\hat{y}[x_i] = \begin{cases} \delta_{1i} & \text{for } i = 1, 2, .., n \\ \delta_{2i} & \text{for } i = n + 1, n + 2, .., 2n \end{cases} \tag{3.26}$$

where $x_i = c\,(i - 1)$, for $i = 1, 2, .., 2n$, and c > 0.

3. Odd indexed $y$ values are obtained from $\delta_{1i}$ and even indexed $y$ values are obtained from $\delta_{2i}$, formally:

$$\hat{y}[x_i] = \begin{cases} \delta_{1i} & \text{for } i = 1, 3, .., 2n - 1 \\ \delta_{2i} & \text{for } i = 2, 4, .., 2n \end{cases} \tag{3.27}$$

where $x_i = c\,(i - 1)$, for $i = 1, 2, .., 2n$, and $\delta = \left(\delta_{1i}, \delta_{2i,}\right)$, and c > 0.

In this dissertation, $\delta_{1i}$ and $\delta_{2i}$ denote the radial distance and the edge distance, respectively.

### 3.4.1.3) The Algorithm to obtain NCPR Shape Function

Input: NCPR $P_\mathrm{N}$.

Output: The shape function $y[x]$.

1. Using equations (3.25) - (3.27), obtain the shape function $\tilde{y}[x]$ that is obtained from $\frac{d\hat{y}[x]}{dx}$ or by changing sign of even indexed $\hat{y}$ values, e.g. $\hat{y}[2] = -\hat{y}[2]$ and so on, or from combination of derivative and sign change.
2. Apply linear interpolation to $\tilde{y}$ to obtain $y[x]$.

In this dissertation, we applied sign change and then derivative to the function defined in (3.26), $c = 1/(2n-1)$, interpolation interval is 0.01, linear interpolation is applied, and first 50 Fourier coefficients are used.

#### 3.4.1.4) The Feature Vector

A feature vector is an abstract representation in a vector form. In this dissertation, the Fourier coefficients are used as a feature vector due to its discriminative property. The feature vector **f** is defined as

$$\mathbf{f} = (\mathbf{Y}[i] | i = 1, 2, .., K) \tag{3.28}$$

where **Y** is the discrete Fourier transform of $y[x]$ and $K$ is the size of **f**.

### 3.4.2) Second Group Shape Features

Elongation is calculated from an ellipse which is fitted to the boundary points of the ROI [Halif00, Hansel]; it is defined as

$$ELN = \frac{\text{length of min axis of the ellipse fitted}}{\text{length of max axis of the ellipse fitted}} \tag{3.29}$$

Eccentricity is defined as

$$ECC = \sqrt{1 - \text{Elongation}^2} \tag{3.30}$$

Circularity by radial distance is defined as

$$CRC_{\text{RD}} = 1 - STD\left(\frac{r_i}{\max\limits_{i=1,2,..,n}(r_i)}\right) \tag{3.31}$$

Circularity by CPR is a global similarity measure between $P_{\text{CPR}}$ (CPR of the ROI's polygon $P$) and $P_{\text{E\_CPR}}$ (CPR of an equilateral polygon $P_{\text{E}}$) which has the same number of vertices with $P$ whose vertices are on a unit circle; it is defined as

$$CRC_{\text{CPR}} = 1 - \frac{D_\alpha + D_\text{d}}{2} \tag{3.32}$$

where $D_\alpha$ and $D_\text{d}$ (which are $[0,1]^n \rightarrow [0,1]$) that are defined by (3.33) are distances between $P_{\text{CPR}}$ and $P_{\text{E\_CPR}}$ (given by (3.38)) $\alpha \in [0°, 90°]$ that is defined by (3.36) is the angle in degrees, $\|*\|$ refers to Euclidean norm of a vector, and $|*|$ refers to an absolute value. The distances are

$$D_\text{d} = \frac{|\,\|Q\| - \|R\|\,|}{\|Q\| + \|R\|} \tag{3.33}$$

$$\|Q\| = \sqrt{q_1^2 + q_2^2}$$

$$\|R\| = \sqrt{r_1^2 + r_2^2}$$

where *Q* and *R* are given by

$$Q = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} \sqrt{\sum_{i=1}^{n} (\delta_{1i})^2} \\ \sqrt{\sum_{i=1}^{n} (\delta_{2i})^2} \end{pmatrix} \tag{3.34}$$

where $\delta_{1i}$ and $\delta_{2i}$ are distance vector $\delta_i$ elements of $P_{\mathrm{CPR}}$, and similarly

$$R = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} = \begin{pmatrix} \sqrt{\sum_{i=1}^{n} (\delta_{1i})^2} \\ \sqrt{\sum_{i=1}^{n} (\delta_{2i})^2} \end{pmatrix} \tag{3.35}$$

where $\delta_{1i}$ and $\delta_{2i}$ are distance vector $\delta_i$ elements of $P_{\mathrm{E\_CPR}}$.

$$\mathrm{D}_\alpha = \frac{\alpha}{90}$$

$$\alpha = \frac{\sum_{i=1}^{2} q_i \, r_i}{\sqrt{\sum_{i=1}^{2} (q_i)^2} \, \sqrt{\sum_{i=1}^{2} (r_i)^2}} \tag{3.36}$$

CPR of an equilateral polygon $P_{\mathrm{E}}$ of *n* vertices is computed by equation (3.37). Let $\delta_{1i}$, $\delta_{2i}$, and $\delta_{2,i+1}$ denote edge length of an equilateral triangle whose corners are located at the centroid and successive two vertices $V_i$ and $V_{i+1}$ of the polygon $P_{\mathrm{E}}$ where $\delta_{2i}$ and $\delta_{2,i+1}$ are radial distance from the center of the polygon to the vertices $V_i$ and $V_{i+1}$, respectively, and $\delta_{1i}$ is the edge distance between two vertices; notice that $r = \delta_{2i} = \delta_{2j} = 1$ and $d = \delta_{1i} = \delta_{1j}$ for *i*, *j* = 1, 2, .., *n* . Let $\theta$ denotes the angle between two radial edges. Using the cosine theorem

$$d^2 = \delta_{2i}^2 + \delta_{2,i+1}^2 - 2\,\delta_{2i}\,\delta_{2,i+1}\,\cos\theta$$

$$d^2 = 2\,r^2(1 - \cos\theta) \tag{3.37}$$

$$d = \sqrt{2(1 - \cos\theta)}$$

Using equation (3.4), we get

$$P_{\mathrm{E\_CPR}} = \begin{pmatrix} \delta_{21} & & \delta_{2n} \\ \delta_{11} & & \delta_{1n} \end{pmatrix} = \begin{pmatrix} 1 & \cdots & 1 \\ d & \cdots & d \end{pmatrix} \tag{3.38}$$

Circularity by Spectral Feature measures the similarity between the equilateral polygon $P_{\mathrm{E}}$ and the polygon *P* using their spectral feature $\mathbf{f}_{\mathrm{E}}$ and **f**, respectively.

$$CRC_{\mathrm{SP}} = 1 - \frac{D_\alpha + D_{\mathrm{d}}}{2} \tag{3.39}$$

where $D_\alpha$ and $D_\mathrm{d}$ are distances between $\mathbf{f}_\mathrm{E}$ and $\mathbf{f}$; $D_\alpha$ and $D_\mathrm{d}$ are computed by (3.33) and (3.36) substituting $Q = \mathbf{f}$ and $R = \mathbf{f}_\mathrm{E}$.

The normalized Euler's number measures how much a ROI is composed of several sub regions. It is defined as

$$NEN = \begin{cases} 1.00 & EN \leq -4 \\ 0.75 & EN = -3 \\ 0.50 & EN = -2 \\ 0.25 & EN = -1 \\ \varepsilon & \mathrm{o.w.} \end{cases} \tag{3.40}$$

where $EN$ is the Euler's number and $\varepsilon$ is a very small positive number; the Euler's number defined as the difference between the number of contiguous regions and the number of holes in the imject.

Compactness gives an idea about the smoothness of the boundary; it is given by

$$CMP = \frac{\text{Area of the ROI}}{(\text{Perimeter of the ROI})^2} \tag{3.41}$$

Convexity is given by

$$CNV = \frac{\text{Area of the ROI}}{\text{Area of convex hull of the boundary of the ROI}} \tag{3.42}$$

### 3.4.3) Photometric Features

Gray level associations conveyed by an imject can be described by photometric features. The main objective of these features is to define some textural variations within a region so that imjects can be discriminated by means of these features. An image region can be described based on its texture coarseness [Haralic79]. In this dissertation, we utilized two textural features: 1) co-occurrence features and 2) run-length features.

Co-occurrence features are obtained from the co-occurrence matrix of the imject. Co-occurrence matrix is an $N$ by $N$ dimensional array whose elements are the number of pixels with the same gray values $g \in G$ within a neighbor specified [Sonka98 p.651]. The neighbor can be defined by a radius and an angle; in this dissertation, we used a rectangular disk neighbor. After computation of the matrix, all elements are divided by the sum of elements to obtain probability matrix $P$. Given a co-occurrence probability matrix $P$ of $N$ by $N$ and a gray-level set $a, b \in G$ which is obtained from the imject, the co-occurrence features used in this dissertation are defined below [Sonka98 p. 651, Haralic79].

Energy: The higher the value of the energy is obtained, the higher the homogeneity exists; it is defined by

$$ENR = \sum_{a}^{N} \sum_{b}^{N} (P(a,b))^2 \tag{3.43}$$

Entropy: The higher the in-homogeneity exists, the higher the entropy is obtained. Since entropy gives a value in the interval $[0, \log_2 N^2]$, we normalized it; the normalized entropy is defined by

$$ENT_{\text{Normalized}} = -\frac{\sum_a^N \sum_b^N P(a,b) \log_2 P(a,b)}{2 \log_2 N} \tag{3.44}$$

Contrast: The contrast gives expected differences among all gray levels; so it measures gray level variations. The normalized contrasts are defined by

$$CNT = \sum_{a \in [g_{min}, g_{max}]} \sum_{b \in [g_{min}, g_{max}]} (a-b)^2 \, P(a,b) \; = \mathrm{E}[(a-b)^2]$$

$$CNT_{\text{NormMax}} = \frac{CNT}{\max\limits_{a,b \in [g_{min}, g_{max}]} [(a-b)^2]} = \frac{CNT}{(g_{min} - g_{max})^2} \tag{3.45}$$

$$CNT_{\text{NormSum}} = \frac{Cnt}{\sum_{a \in [1,K]} \sum_{b \in [1,K]} (a-b)^2}$$

The Inverse difference moment: It gives expected reciprocal of differences among all gray levels; it is given by

$$IDM = \sum_a \sum_b \frac{P(a,b)}{(a-b)^2} = \mathrm{E}\left[\frac{1}{(a-b)^2}\right], a \neq b \tag{3.46}$$

The Correlation among the gray levels: The higher the correlation is obtained, the higher the homogeneity exists. It is defined as

$$COR = \frac{\sum_{a \in G} \sum_{b \in G} a \, b \, P(a,b) - \mu_{\mathrm{A}} \, \mu_{\mathrm{B}}}{\sigma_{\mathrm{A}} \, \sigma_{\mathrm{B}}} \tag{3.47}$$

where

$$\mu_{\mathrm{A}} = \sum_a a \sum_b P(a,b), \mu_{\mathrm{B}} = \sum_b b \sum_a P(a,b)$$
$$\sigma_{\mathrm{A}}^2 = \sum_a (a - \mu_{\mathrm{A}})^2 \sum_b P(a,b), \sigma_{\mathrm{B}}^2 = \sum_b (b - \mu_{\mathrm{B}})^2 \sum_a P(a,b) \tag{3.48}$$

Run-length features of an imject are obtained from its run-length array. The run-length array *B* has two dimensions, namely gray-levels *G* and run-lengths *R*. An element *B*(*g*, *r*) of the array is the number pixels which has *r* max connected gray-level *g* pixels within a neighbor specified. Given a run-length matrix *B* of *M* by *N*, the run-length features are defined below.

$$K = \sum_g^M \sum_r^N B(g,r) \tag{3.49}$$

Modified gray-level non-uniformity: We used the quadratic form of the regular non-uniformity and divide run-length values by *r* and *g* to increase the contrast and accordingly the discrimination capability of the measure. It is defined as

$$MGLNU = \frac{1}{K} \sum_g^M \frac{1}{g^2} \left\{ \sum_r^N \frac{B(g,r)}{r} \right\}^2 \tag{3.50}$$

Short-run emphasis is given by

$$SRE = \frac{1}{K}\sum_{g}^{M}\sum_{r}^{N}\frac{B(g,r)}{r^2} \tag{3.51}$$

Long-run emphasis is given by

$$LRE = \frac{1}{K}\sum_{g}^{M}\sum_{r}^{N}B(g,r)\,r^2 \tag{3.52}$$

Run percentage is given by

$$RP = \frac{K}{M\,N} \tag{3.53}$$

Modified run-length non-uniformity is given by

$$MRLNU = \frac{1}{K}\sum_{r}^{N}\frac{1}{r^2}\left\{\sum_{g}^{M}\frac{B(g,r)}{g}\right\}^2 \tag{3.54}$$

In calculation of the run-length non-uniformity, we used the quadratic form based on the reason explained in gray-level non-uniformity.

# CHAPTER 4. DATA PREPARATION

The raw data needs to be pre-processed before obtaining feature vector set for training and testing. This preprocessing includes 1) creation of VOI from slice series, 2) scale normalization, 3) slice selection, 4) smoothing, and 5) obtaining polygonal representation of the boundaries.

The nodule and non-nodule image series are provided by National Cancer Institute from National Cancer Imaging Archive [NCIA] and by Lake P.E.T. Imaging Center [LakePet], respectively. The nodules were marked by four different radiologists and their location is saved as an XML file. The non-nodule imjects are obtained semi-automatically using the tool SHAHIN developed; its user interface is seen in the Figure 4.1. In this research, we used 80 nodules and non-nodules imjects in total with 40 imjects from each class. Image series has a resolution of 512 by 512 resolution and slice thickness from 2 to 5 mm.

Collection Strategies of Samples: Samples to be used in training and testing can be collected in several ways depending on how to design feature type predictors that is used in model layer 1. A classifier can be designed to analyze a certain slice only (say 1st slice of the transverse plane), a certain plane only, or a certain feature regardless of slice/plane location. Hence samples can be collected by any of the three methods as summarized in the Table 4.1.

Table 4.1 Collection strategies of samples (using 3 planes, *S* slices per plane, *K* features per slice)

| Collection Strategy | To Analyze | Samples From | Sets Of Samples | # Classifiers |
|---|---|---|---|---|
| Slice-specific | slicess {1, 2, .., *S*} | slices {1, 2, .., *S*} | 3 *S* | 3 *S K* |
| Plane-specific | planes {1, 2, 3} | all slices from the plane | 3 | 3 *K* |
| Feature-specific | features {1, 2, .., *K*} | all slices from each plane | 1 | *K* |

Slice-specific sample collection: Each classifier is assigned to a specific slice of the plane. The training set is composed of the feature vectors obtained from the specific slice location, say from 1st slice of the first plane. Hence a classifier is specific to a certain slice. As an example, if 3 slices per plane and 10 features are used, then there will be 9 sets of samples namely samples from plane 1 slice 1, plane 1 slice 2, and so on; therefore, the total number of classifiers will be equal to the number of slices times the number of planes times the number of features, which is 90 in total.

Plane-specific sample collection: Each classifier is assigned for analyzing a specific plane, e.g. X-Y plane. The training set is a composition of the feature vectors obtained from all slices in a specific plane. Hence a classifier is specific to a certain plane. There will be 3 sets of samples and therefore the total number of classifiers will be equal to the number of planes times the number of features, which is 30 in total.

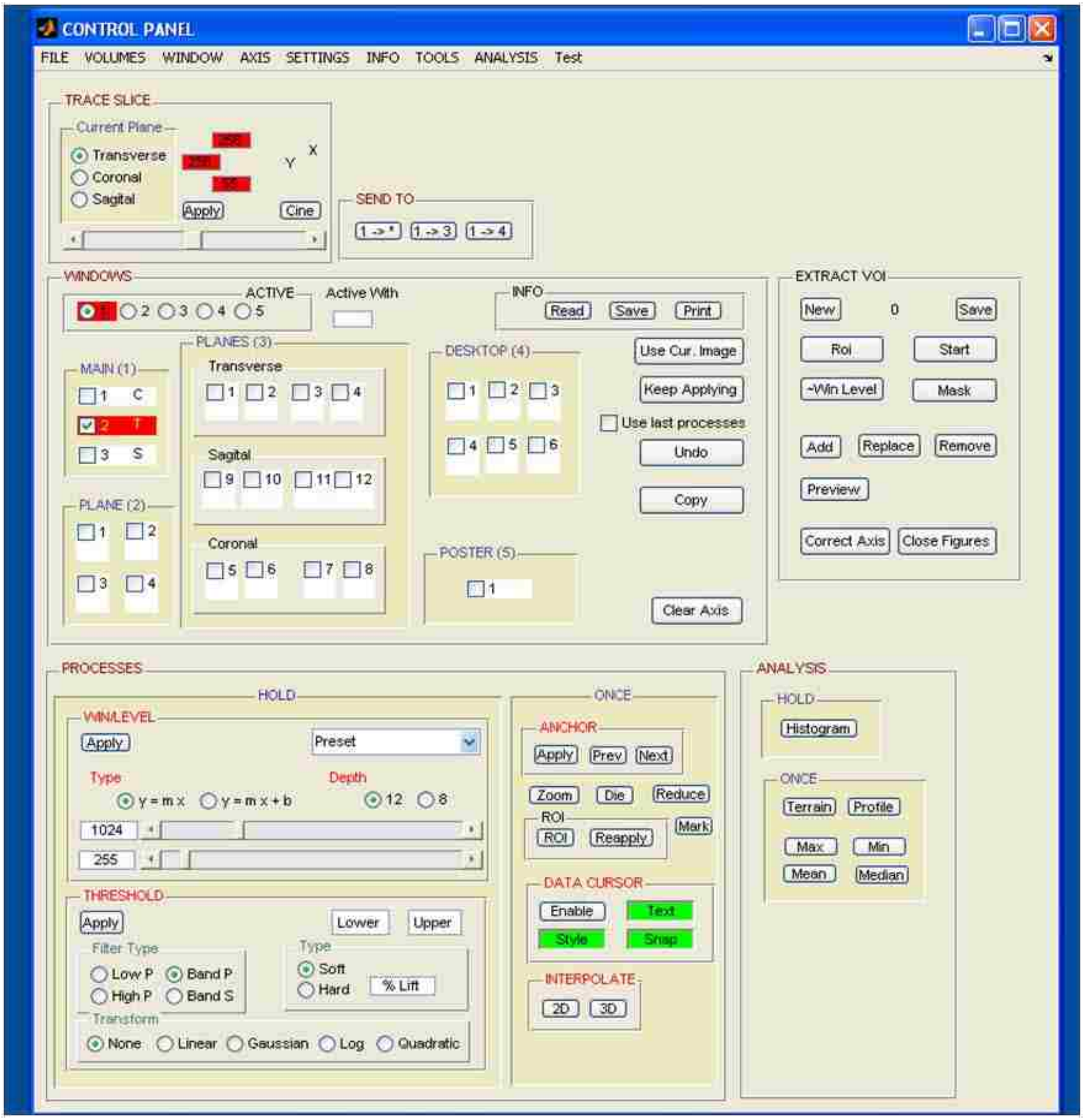


Figure 4.1 The interface for visualization and feature extraction tool

Feature-specific sample collection: Each classifier is assigned for analyzing a specific feature regardless of the slice location in the VOI. The training set is a composition of the feature vectors obtained from all slices in the VOI. Hence a classifier is specific to a feature. Therefore, the total number of classifiers will be equal to the number of features which is 10 in total.

Retrieval of VOI: We developed a tool to retrieve the VOI outlines from the XML file which stores information about nodule image series; its process diagram is shown in Figure 4.3. Since image series are stored in different ranges, images are shifted to the gray level range starting from 1 for foreground pixels while making background pixels as zero; this also relieves the photometric feature extraction. The coordinate system of the images and the software (MATLAB) are synchronized. A mask is placed around ROI to filter out the rest of the image pixels. Then slices are stacked to obtain the whole VOI. Non-nodule imjects are extracted by the author of this dissertation using the tool developed from a patient image series that does not have any nodules in it.

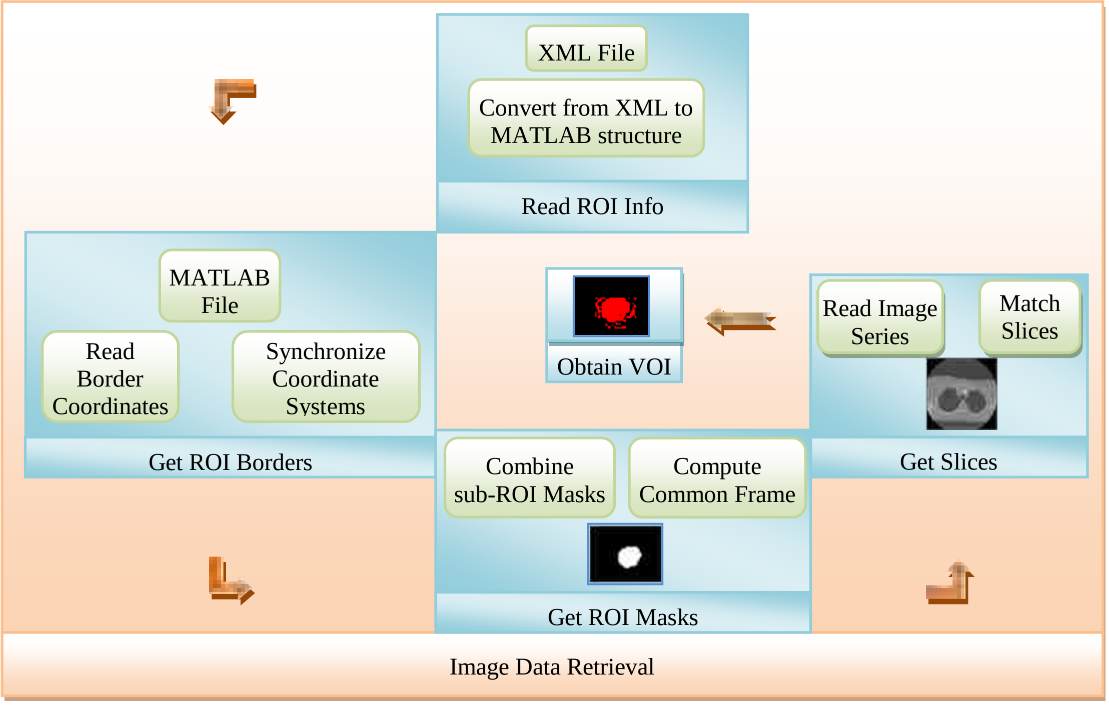


Figure 4.2 Image data retrieval unit of the framework

## 4.1) Preprocessing of VOIs

Since all of the VOIs are not at the same size and also some slices may have a ROI which does not have enough pixels to extract some features, scale normalization and slice selection processes are applied before the feature extraction.

VOI Normalization: Image voxels are not isotropic and they differ among image series. An interpolation (e.g. trilinear or nearest neighbor) is applied to each VOI to obtain isotropic voxels; the interpolation helps to make all VOI from each plane have the same number of slices. This normalization is necessary to make each sample group has the same number of slices; because the size of feature vectors at model layer 2 requires a fix number of predictions as an input.

Slice Selection: Since a VOI may not have a regular shape, a series of slices through a plane direction may have 2D regions whose size is less than 9 pixels. This type of slices cannot be used to extract feature vectors. They are replaced by the nearest slice whose ROI has an area of more than 9 pixels. If a VOI has a slice on which more than one disconnected regions exist, then the region which has the max area is automatically selected while deleting others. After finding the max region, its boundary is smoothed for shape feature extraction. The algorithm to select a slice is given below.

Input: 2D image with background pixels

Output: 2D image with max region

```
1. Find all regions
    a. Trace the image line by line from the first pixel to the last
        i. Find the first non-background pixel
        ii. Label each non-background neighbor pixel with the same value
        iii. If a non-background neighbor pixel does not have the same label, correct the labels.
2. Compute the area of each region
3. Delete all regions whose area less than the max area
4. Apply smoothing to the boundary of max area
5. Re-obtain the ROI from the smoothed boundary
6. Return the slice
```

### 4.1.1) Auto-Polygonalization

The boundary of a ROI and its polygonal approximation are obtained automatically. The polygon is used to compute shape features. The algorithm requires five parameters, namely min length for horizontal, for vertical, and for diagonal segments and a tolerance angle to decide if two successive line segments are inline. The algorithm auto-polygonize is given below.

Input: 2D image with background pixels, parameters of min segment lengths and inline tolerance angle

Output: Polygon of the ROI boundary

```
1. Trace the boundary from the first pixel to the last in counter clock direction
    a. Obtain a line segment
    b. Check if an inline condition occurs between the new line segment and the previous one
    c. If the neighbor line segments are inline, then combine them to obtain one line segment.
        i. Successively trace back to eliminate any line segment which becomes inline due to combining neighbor line segments
2. If the size of the polygon less than a specified value, then decrease the parameter values and re-run the step 1.
3. Return the polygon vertices.
```

## 4.2) Features

After obtaining set of ROIs, the next step is to extract feature vectors which are utilized to train and test the decision engine. In this dissertation, we used two feature classes each having two feature types, namely FC = {Geometric, Photometric} and FT = {{Spectral Feature, Circularity}, {Co-Occurrence, Run-length}}, and two object class C = {Nodule, Non-Nodule}; notice that $\mathrm{F_T}(1,1) =$ Spectral feature, $\mathrm{F_T}(2,1) =$ Co-Occurrence feature and so on.

Two types of shape features, namely spectral feature and circularity combined with some regional shape features, are used. The size of the spectral feature vector is selected as 50 which half of the Fourier coefficients computed as described in section (3.4.1.1). The second feature type is used to measure circularity of a ROI along with some regional features. Its feature vector is composed of 5 circularity measures including elongation, eccentricity, circularity by radial distance, circularity by CPR, circularity by spectral feature, and 3 regional measures including normalized Euler's number, compactness, and convexity.

Photometric features are computation starts with binning (with a bin size of 10) over the gray level range so that the matrix dimension is reduced significantly. The gray level range is selected as [1, 1200] because the number of occurrences of gray levels above this range is not significant as seen in Figure 4.3; notice that gray levels of original image slices are shifted to 1 as mentioned earlier. We used a 'disk neighbor' which includes all pixels within a rectangular 3 by 3 window. The disk neighbor enables a rotation invariance property. The direction of run-length trace is selected as {0, -45, -90, -135} so that a previously visited pixel at the same line is not counted more than once.

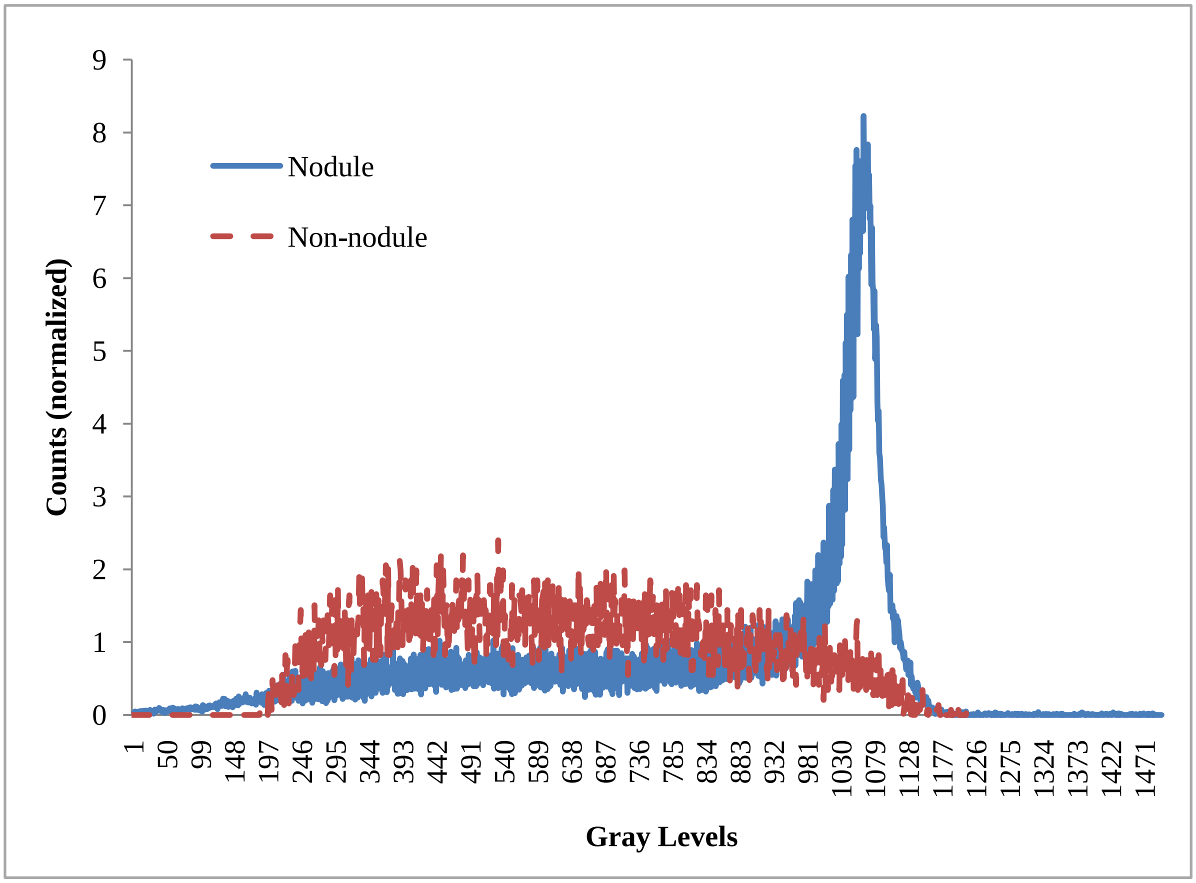


Figure 4.3 Gray levels histogram of imjects used (normalization factor: 210.5 for nodule, 14.6 for non-nodule counts)

## 4.3) A Discussion Related to Invariance Issue

One of the important challenges in pattern recognition is that imjects to be classified differ in size, rotation, and location. In general, Machine Vision may require rigid body invariance in features used to represent patterns; that is the translation, rotation, and scale invariance properties may be necessary. As regard to the nodule detection, the location information may help; so, relative location information also can be helpful as an attribute. But this location information does not need to be reflected in a shape or photometric feature; so these features can be represented removing spatial-origin information of the pattern location. Also the same argument is valid for the orientation of the imject; as regard to the shape representation, in general, orientation information is removed before extraction of the shape feature (rotation invariance). On the other hand, orientation information may be important to represent an imject using photometric content as in co-occurrence or run-length features. The third property of the rigid body transformation (or deformation) is the scale invariance. Shapes that are similar under rigid-body deformation (that is parallel lines remain parallel preserving aspect ratios after removing location, orientation, and scale information) can be grouped in the same cluster; if an application requires representing all shapes in the same cluster, then a single feature will be enough to represent all of these shapes. On the other hand, if the application requires preserving scale information, then each shape should be represented with its own feature. Another shape similarity clustering would be normalizing each shape on a square (or on a cube for 3D volumes) and extracting a single feature for all similar shapes after removing translation and orientation information. In Figure 4.4, three shape clusters are given; a single feature is enough to represent the middle cluster shapes; for the left cluster, each shape may require its own feature or all shapes can be represented by the feature derived from the normalized shape (located at the center); the shapes in the right cluster have the same number of vertices but their shape feature will different. In this dissertation, we derived shape features after applying 'cube normalization' by interpolating each VOI to transform them into 50 by 50 by 50 cube in pixels.

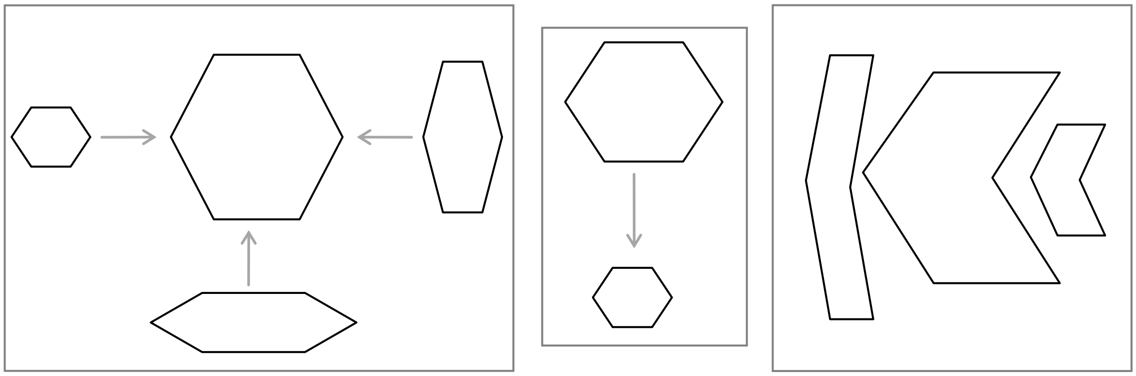


Figure 4.4 Scale invariance over shape clusters. Left cluster: Square (or cube) normalization, middle cluster: rigid body normalization, right cluster: A different cluster of shapes.

# CHAPTER 5. MODULAR DECISION ENGINE MODEL

We propose three different modular decision engine models: Inter-Slice Inter-Plane (ISIP) Model, Inter-Slice Intra Plane (ISAP) Model, and Intra-Slice Intra-Plane (ASAP) Model. In this dissertation, we implemented the first model and left the other two as future research. The ISIP model analyzes a volume based on each feature-type separately and then synthesizes these individual feature-type predictions for each feature-class separately before reaching the final classification; i.e., we are first looking for the answer to a question like "What is class of the imject in the VOI according to the spectral feature?" to be used as a basis prediction. The ISAP model analyzes each plane separately based on all feature-types, and then synthesizes predictions obtained from this analysis to be used in plane level prediction, after the plane level synthesization the final classification is obtained; hence we seek first an answer to a question like "What is class of the imject in the VOI according to all feature-types when we look through a plane?" before further predictions. The ASAP model analyzes each slice based on all features followed by feature-class predictions separately, and then plane level predictions are obtained from these slice level predictions before the final classification at the volume level.

The each proposed decision engine model is a hierarchical engine which has several tiers. The engine analyzes a volumetric data based on selected features by obtaining some intermediate decisions from the first layer to the final layer. The intuition behind the proposed models is improving accuracy of a prediction through hierarchically connected information levels. Our experimental results show that this multi-level approach has a capability of obtaining the most improved decision at the final level. The engine is composed of several layers which contain several modules of a predictor and a regrouping processor. Each module at the same model layer is totally independent from each other while they are integrated to some modules at the successive layers. This independent processing makes it possible to implement the decision engine in a parallel processing environment.

## 5.1) The Data Model of the Modular Decision Engine

The decision engine analyzes a volumetric data at different information levels. These levels include spatial levels as slice, plane, and volume and feature levels as feature-class and feature-type. Spatial information levels are obtained by partitioning 3D space into three orthogonal planes[♣] — namely transverse (*x*-*y*), coronal (*x*-*z*), and sagittal (*y*-*z*) — and the data in the direction of each plane is sub-divided into 2D slice sequences which are composed by the most primitive unit (voxel). At the slice level, a prediction is obtained independently for each slice; at the plane level, predictions are performed for all slices of the plane. Similarly, the volume level synthesization utilizes the plane level outputs to reach a higher level decision. Feature-class (or feature-type) information level synthesization is performed using

[♣] As a convention, the term 'slice' alone refers to 'spatial slice' on a plane and the term 'plane' alone refers to 'spatial plane' in 3D Cartesian coordinate system.

prediction results of other information levels by employing a specific feature-class (or feature-type) predictor. In a non-modular predictor, all of the volumetric data is utilized together to obtain a decision. In contrast, the modular approach partitions the raw data into several spatial and feature levels and combines a hierarchy of predictions at these levels in an efficient way, as explained below. Therefore, the proposed modular engine model mimics a human expert who analyzes a VOI from different projections; further, the engine, after obtaining machine coded features from the raw data, enhances the decision-making process with multi-level feature analysis which is beyond the limit of a human intelligence.

The raw data used by the modular engine is a region as opposed to a non-modular classifier which utilizes all voxels within a volume. Formally, let a voxel $v_{ms}^{(p)}$ on a slice *s* of a planar direction *p* (located at a grid point *m* of *s*) where *s* = 1, 2, .., $N_{\mathrm{S}}^{(p)}$ and *m* = 1, 2, .., $N_{\mathrm{V}}^{(p)}$ where $N_{\mathrm{S}}^{(p)}$ denotes the number of slices per plane and $N_{\mathrm{V}}^{(p)}$ denotes the number of voxels per slice; when Cartesian coordinate system is used, *p* = 1, 2, 3 corresponds to transverse, coronal, and sagittal plane's direction. We define a region as $R_s^{(p)} = \left(v_{1s}^{(p)}, v_{2s}^{(p)}, .., v_{N_{\mathrm{V}}^{(p)}s}^{(p)}\right)$; accordingly a volume in the perspective of plane *p* can be represented by a sequence of voxels $V^{(p)} = \left(v_{11}^{(p)}, v_{21}^{(p)}, .., v_{N_V^{(p)}1}^{(p)}, .., v_{1N_S^{(p)}}^{(p)}, .., v_{N_V^{(p)}N_S^{(p)}}^{(p)}\right)$ or by a sequence of regions $V^{(p)} = \left(R_1^{(p)}, R_2^{(p)}, .., R_{N_{\mathrm{S}}^{(p)}}^{(p)}\right)$. Hence, input feature vector of a modular decision engine will be $\mathbf{I} = f\left(\left(R_1^{(p)}, R_2^{(p)}, .., R_{N_{\mathrm{S}}^{(p)}}^{(p)}\right)\right)$ while $\mathbf{I} = g\left(\left(v_{11}^{(p)}, v_{21}^{(p)}, .., v_{N_V^{(p)}1}^{(p)}, .., v_{1N_S^{(p)}}^{(p)}, .., v_{N_V^{(p)}N_S^{(p)}}^{(p)}\right)\right)$ for the non-modular classifier where *f*(·) and *g*(·) are functions from the regions and the voxels to a feature space, respectively. Figure 5.1 illustrates a non-modular classifier with a volumetric data as an input. In this dissertation, we use region-based representation in such a way that each slice has one region.

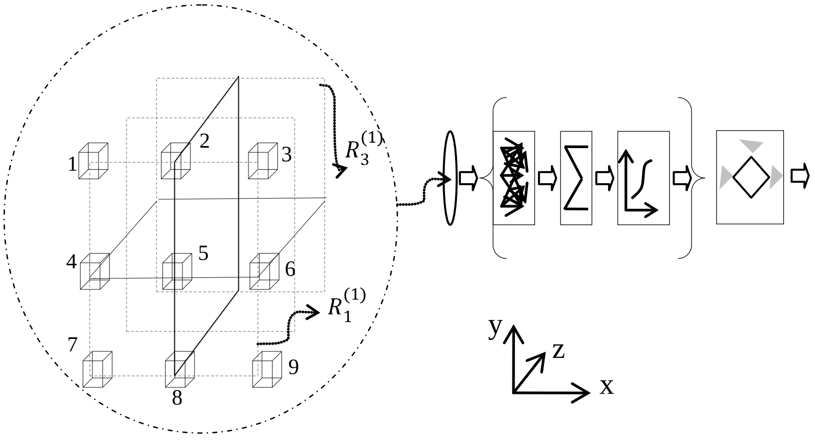


Figure 5.1 A non-modular classifier whose inputs are voxels of a specified plane (Number of voxels $N_{\mathrm{V}}^{(1)} = 9$, number of slices $N_{\mathrm{S}}^{(1)} = 3$ on the plane *p* = 1)

## 5.2) The Modular Decision Engine Structure

The engine is composed of several integrated layers. Each layer has some modules which output a prediction to be used in the next layer. A module contains a predictor and a regrouping processor. The predictor can be any type of processor which employs a supervised (or unsupervised) learning algorithm. All predictors in the engine produce an output in continuous domain; at the last layer a threshold is applied to obtain the final decision. On the other hand, predictors can be designed to produce an intermediate decision in discrete form, e.g., using {0, 1} in case of binary classification, which represents predicted class of the input vector. In this dissertation, we implemented the former approach. The regrouping processor transforms the set of output responses of a predictor into an input vector for the modules at the next layer.

### 5.2.1) Predictor Design

A predictor can be any supervised or un-supervised learning model such as ANN, Support Vector Machine, Bayesian Network model, a multi-dimensional regression model, or a rule based system. In this dissertation, we employed ANN with supervised learning algorithm Backpropagation. A predictor is composed of input, generalization, and output units. Input unit accepts set of vectors and feed them to the generalization unit to adjust weights, then generalization unit's response is compared with the target; finally, the output unit displays the result with an appropriate coding. A spatial information level predictor is trained in such a way that it can make a prediction based on an input feature vector regardless of its spatial origin. In other words, domain of a predictor spans input vector space of all planes and slices; hence the same predictor is utilized for a single feature-type.

A predictor can be designed to make analysis at the slice level using a supervised (and un-supervised) learning method. Also, an un-supervised learning method can be employed following a supervised predictor unit. This approach can assist to cluster input feature vectors into some intermediate feature-clusters then the feature level analysis is performed using cluster information of the samples. In this dissertation, we implemented the supervised learning method alone. These two approaches are illustrated in Figure 5.2.

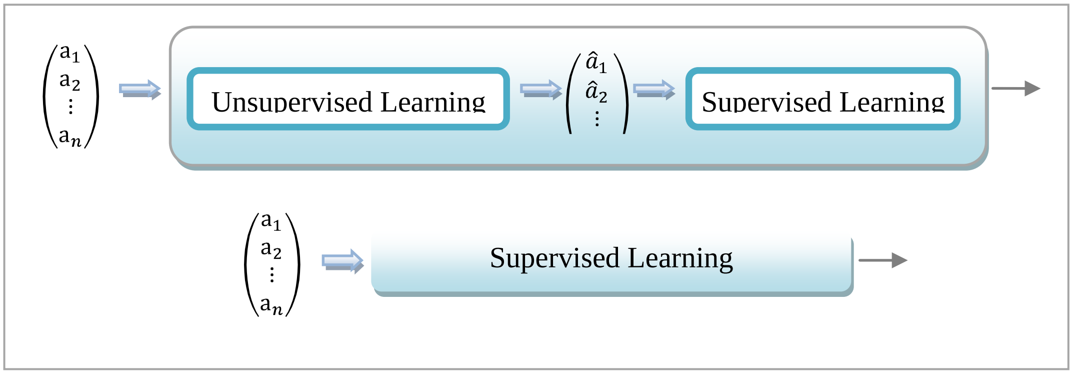

$\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$
Unsupervised Learning
$\begin{pmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \vdots \end{pmatrix}$
Supervised Learning
$\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$
Supervised Learning

Figure 5.2 Prediction methods ($m < n$, $\hat{a}_i$: Feature-cluster value of the input feature-vector for the cluster $i$)

### 5.2.2) Regrouping Processor Design

Let $N^{(l)}$ and $N^{(l)}$ be the size of an output set and of an input vector, respectively. The regrouping processor's ($\mathrm{RGP}_l$) job is to transform a 1 by $N^{(l)}$ output set at a layer $l$ to a $M^{(l+1)}$ by $N^{(l+1)}$ input vector to be used by the modules at next layers. The size of an input vector and of output set  are determined before training of the engine. The output set can be composed of several groups and each group can be composed of several blocks of elements such as slices. The detail of the processor is explained with an example in the Section (5.3.1).

## 5.3) Inter-Slice Inter-Plane (ISIP) Model

The inter-slice inter-plane decision engine start analyzing a VOI from the slices of a plane to obtain some predictions at the feature-type level. Then, a synthesization occurs at the plane level followed by the volume level synthesization; the overall decision is obtained by the module $\mathrm{M}_{\mathrm{O}}$. Figure 5.3 and Figure 5.4 illustrate ISIP structure of the modular decision engine model with two feature-classes of two feature-types for each. The model layers are explained in the sequel.

Layer 1. Feature-type Analysis of the VOI: This layer is composed of feature-type modules $\mathrm{M}_{\mathrm{F}\bullet}$; e.g., $\mathrm{M}_{\mathrm{F},f=1,FC=1}$ is composed of $\mathrm{P}_{\mathrm{F11}}$ and $\mathrm{RGP}_{\mathrm{l}}$. The first prediction about each slice based on a specific feature-type is obtained at this layer. The decision engine seeks for an answer to a question like "to what extent does the imject in the slice belong to the nodule class according to each individual feature-type?".

Layer 2. Plane and Volume Level Synthesization: This layer is composed of plane level modules $\mathrm{M}_{\mathrm{P}\bullet}$ and volume level modules $\mathrm{M}_{\mathrm{V}\bullet}$; e.g., $\mathrm{M}_{\mathrm{P11}}$ is composed of $\mathrm{P}_{\mathrm{P11}}$ and $\mathrm{RGP}_2$, and $\mathrm{M}_{\mathrm{V11}}$ is composed of $\mathrm{P}_{\mathrm{V11}}$ and $\mathrm{RGP}_3$. An analysis is conducted separately on the VOI based on each individual feature type; first, a prediction is obtained at the plane level, and then output of the 3 plane predictors are used to obtain a volume level prediction. The decision engine seeks for an answer to a question like "to what extent does the imject in the VOI belong to the nodule class according to a shape feature when we look through planes individually first then altogether?". On the other hand, instead of using two types of module (namely plane and volume level modules), plane level prediction can be skipped and only volume level synthesization can be performed; this is depicted as dashed line box ($\mathrm{M}_{\mathrm{P}\bullet}$) in Figure 5.3.

Layer 3. Feature-Class Level Synthesization: This layer is composed of feature-class level modules $\mathrm{M}_{\mathrm{FC}\bullet}$; e.g., $\mathrm{M}_{\mathrm{FC1}}$ is composed of $\mathrm{P}_{\mathrm{FC1}}$ and $\mathrm{RGP}_4$. A prediction about the VOI is obtained based on all features types of the feature-classes' outputs. The decision engine seeks for an answer to a question like "to what extent does the imject in the VOI belong to the nodule class according to predictions considering all geometric features?".

Layer 4. Final Decision: This layer has a module $\mathrm{M}_{\mathrm{O}}$ which produces the final decision about the VOI. The module has a threshold operator after the predictor unit to obtain the final classification result. The decision engine seeks for an answer to a question like "to what extent does the imject in the VOI belong to the nodule class considering geometric and photometric features altogether?".

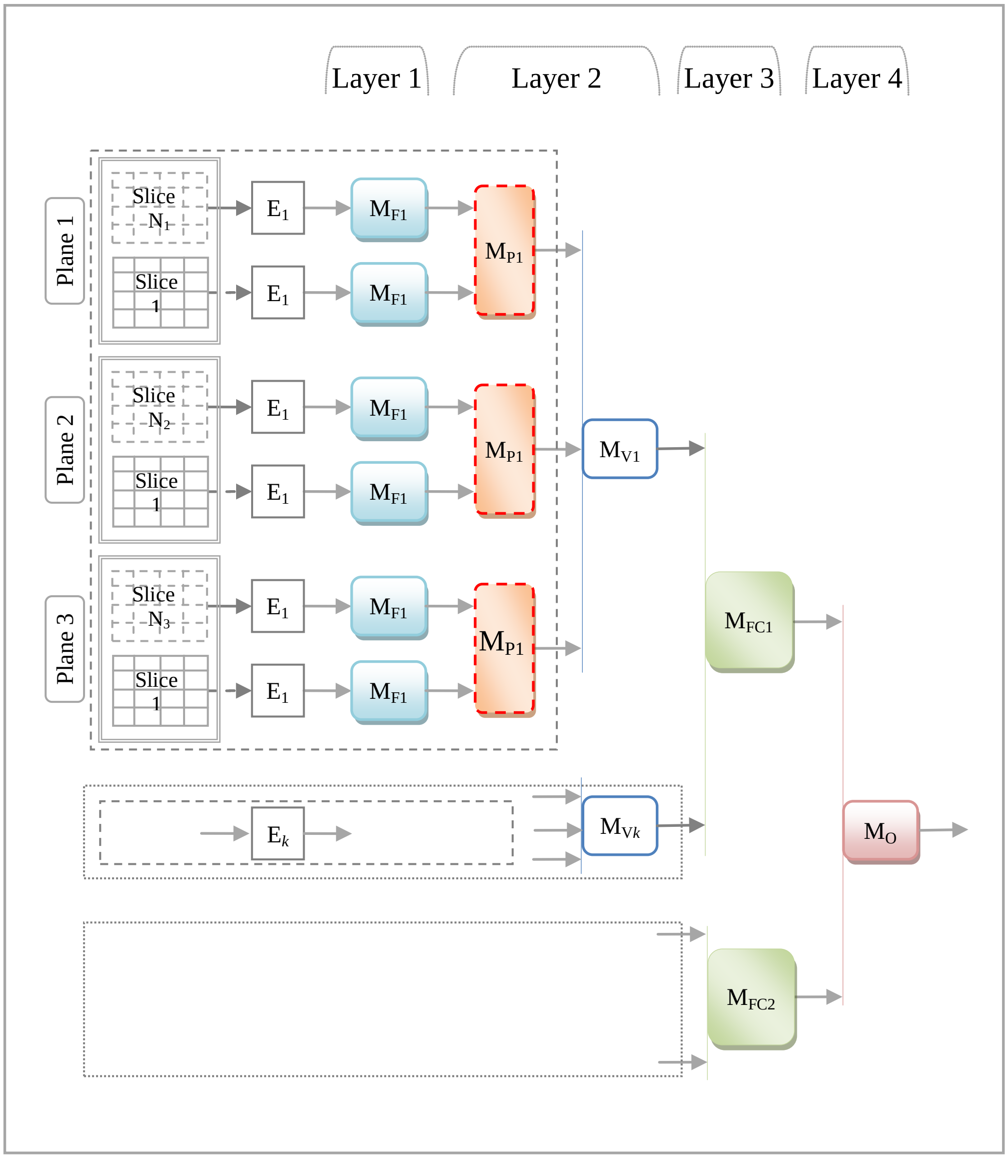


Figure 5.3 The ISIP conceptual model structure ($M_{Fk}$: Feature-type level module, $M_{P\bullet}$: Plane level module, $M_{V\bullet}$: Volume level module, $M_{FC\bullet}$: Feature-class level module, $M_O$: Overall classification module, $E_k$: Feature Extractor of a feature type $k$)

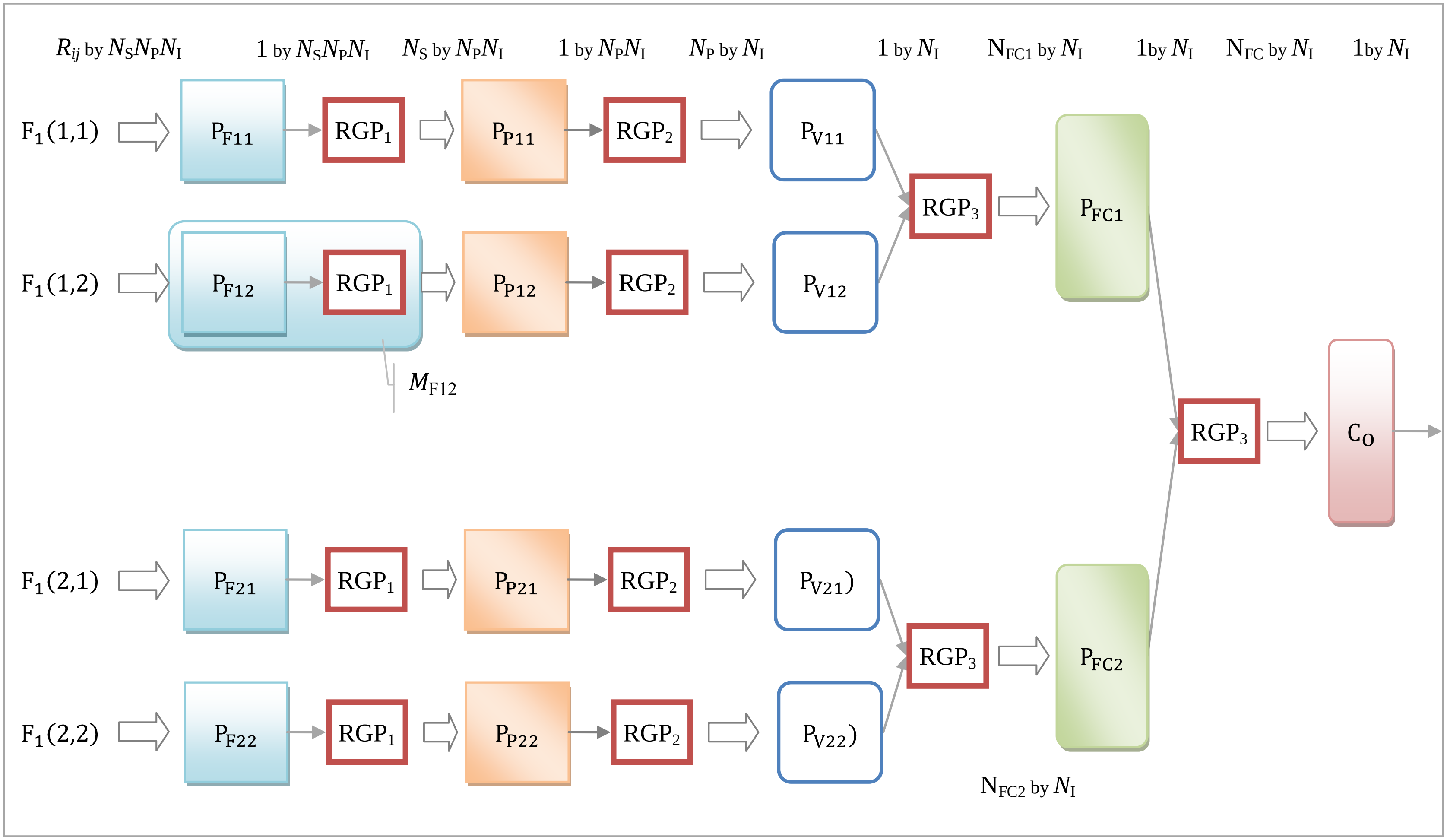


Figure 5.4 An example of ISIP implementation model structure (with two feature-classes and two feature-types in each class. $P_F$: Feature-type level predictor, $P_P$: Plane level predictor, $P_V$: Volume level predictor, $P_{FC}$: Feature-class level predictor, $C_O$: Overall classifier)

### 5.3.1) Regrouping Process

In ISIP model, the regrouping processor takes output set of a module as an input and returns an input vector for the next layer. Let $N_\mathrm{S}$, $N_\mathrm{P}$, and $N_\mathrm{I}$ be the number of slices per plane, the number of planes, and the number of imjects, respectively. At the first model layer, there will be $N_\mathrm{I}$ groups, $N_\mathrm{P}$ blocks in each group, and $N_\mathrm{S}$ elements in each block where the block elements corresponds to slices of a plane, so the size of an output set is $N_\mathrm{S} \cdot N_\mathrm{P} \cdot N_\mathrm{I}$; in this dissertation, $N_\mathrm{S} = 50$, $N_\mathrm{P} = 3$, $N_\mathrm{I} = 80$ (48 for training, 16 for validation, and 16 for testing). $\mathrm{RGP}_{l=1}$ takes the output set and returns $N_\mathrm{P} \cdot N_\mathrm{I}$ input vectors of $N_\mathrm{S}$ attributes in each. At model layer two, the plane decider produces $N_\mathrm{P} \cdot N_\mathrm{I}$ outputs and $\mathrm{RGP}_{l=2}$ returns $N_\mathrm{I}$ input vectors with $N_\mathrm{P}$ attributes in each. The volume, feature-class, and overall classification modules (at layers three, four, and five, respectively) produce $N_\mathrm{I}$ outputs and $\mathrm{RGP}_{l=3}$ returns $N_\mathrm{I}$ input vectors with $N_{i,j}^{(l=3)}$ (= 2 in this dissertation) attributes for a feature-type $\mathrm{F_T}(i, j)$ and $\mathrm{RGP}_{l=4}$ returns $N_i^{(4)}$ (= 2 in this dissertation) attributes for a feature-class $\mathrm{F_C}(i)$. Figure 5.5 and Figure 5.6 illustrate the regrouping process.

Formally, Let $b_s$ denotes a predicted value for a slice $s$, $B_p = \left(b_1, b_2, .., b_{N_\mathrm{S}}\right)$ denotes a block for a plane $p$, $G_k = \left(B_1, B_2, .., B_{N_\mathrm{P}}\right)$ denotes a group for an imject $k$, and $O_l(i,j) = \left(G_1, G_2, .., G_{N_\mathrm{I}}\right)$ denotes a set of output groups obtained from a feature-type predictor $\mathrm{P_F}(i,j)$ at model layer $l$ as a response to the feature vectors of $F_l(i,j)$. A group is made of $N_\mathrm{P}$ blocks where each block corresponds to a plane and the size of each block is $N_\mathrm{S}$. The input vector of the second layer with 3 planes will be

$$F_2(i,j) = ([(B_1)',\ (B_2)',\ (B_3)']_{k=1},\ [(B_1)',\ (B_2)',\ (B_3)']_2, ..) \qquad (5.1)$$

where $(*)'$ denotes the transpose operator. Let a group $G_1 = (B_1, B_2, B_3)$; similarly, the second layer has an output set $O_2(i,j) = (G_1,\ G_2, ..)$ whose blocks ($B_p = b$ for $p$ = 1, 2, 3) correspond to a plane with one element in each as a response to input vectors $F_2(i,j)$. The input prediction vector of the third layer will be

$$F_3(i,j) = ((G_1)',\ (G_2)', ..) \qquad (5.2)$$

Let $o_{jk}$ denotes a prediction for the $k$th imject and $j$th feature type, and $O_3(i,j) = \left(o_{j=1,k=1},\ o_{12}, ..\right)$ denotes a set of outputs of a volume feature predictor $\mathrm{P_V}(i,j)$ at the model layer three. The prediction vectors of the fourth model layer are obtained by combining output of the previous layer classifiers as defined below

$$F_4(i) = ((o_{11}, o_{21}, ..)', (o_{12}, o_{22}, ..)', ..) \qquad (5.3)$$

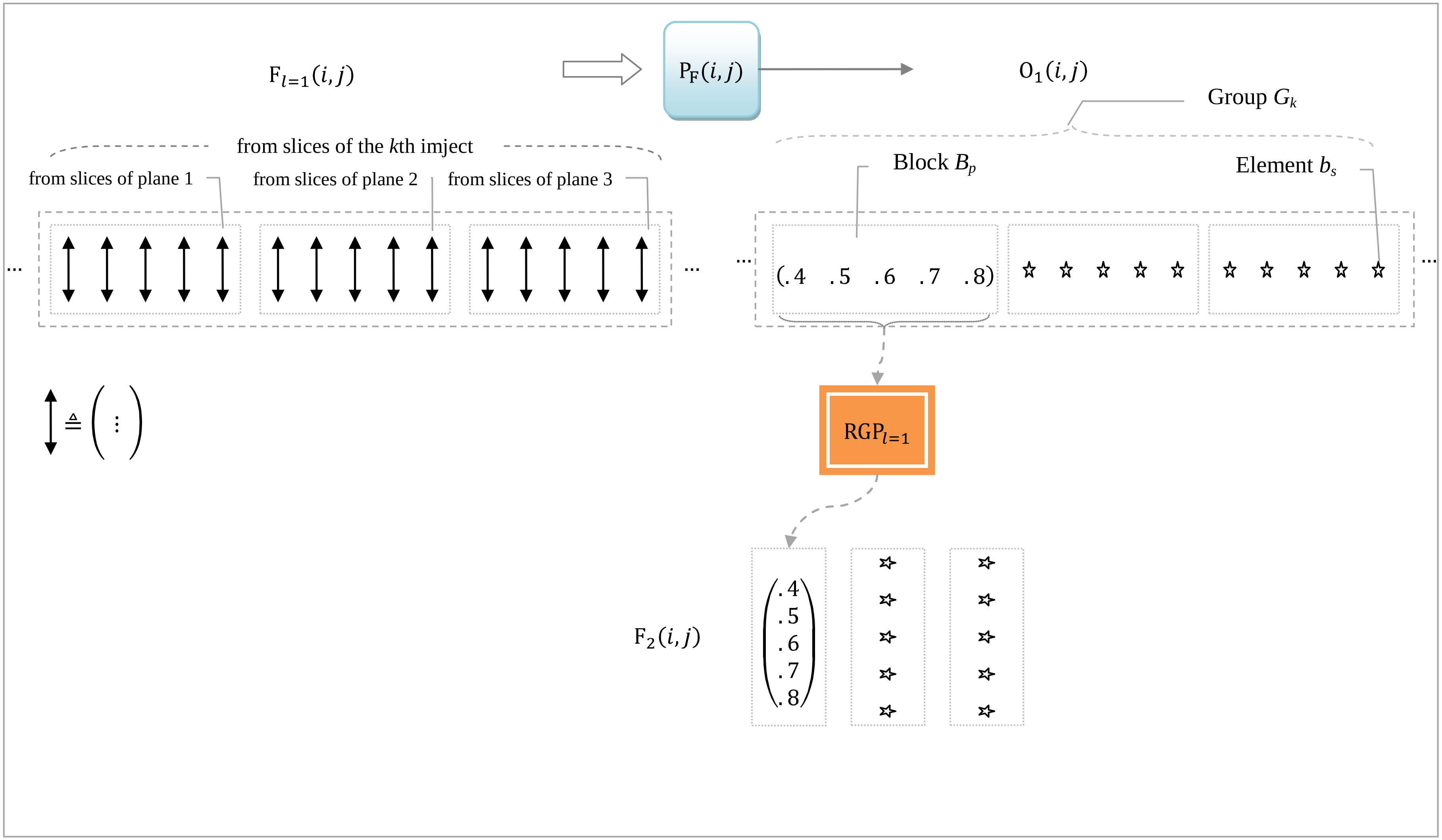


Figure 5.5 Regrouping process at layer 1 (ISIP Model with 5 slices per plane, 3 planes per imject)

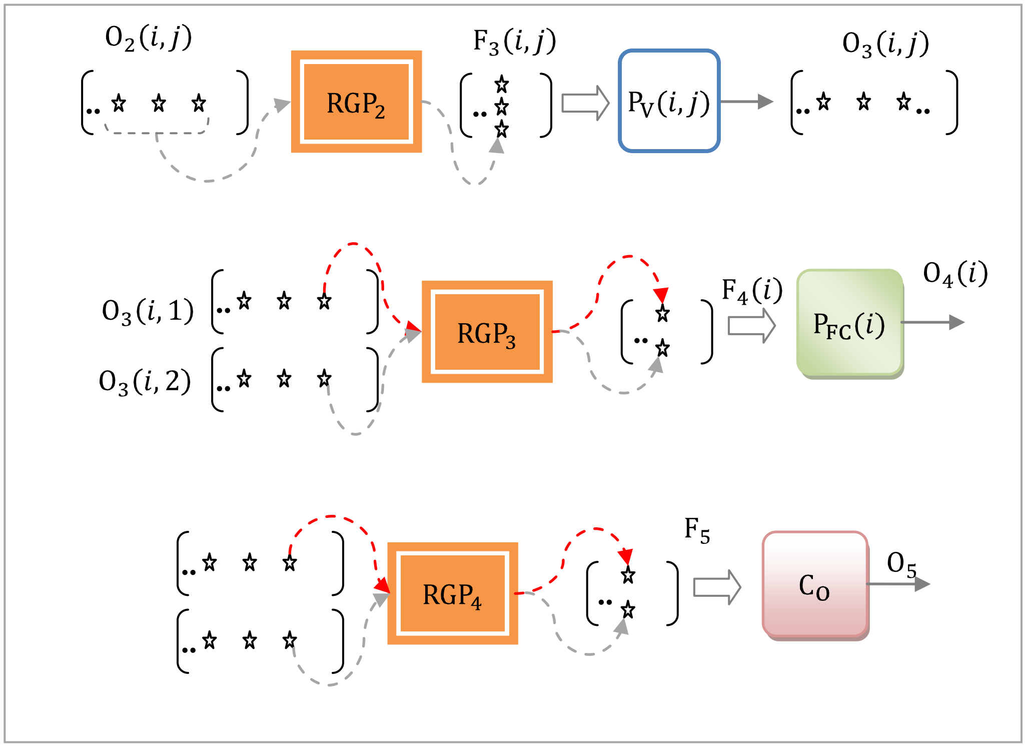


Figure 5.6 Regrouping process at layers 2, 3, and 4 (ISIP Model)

Similarly, let $o_{ik}$ denotes a prediction for the $k$th imject and $i$th feature class, $O_4(i) = (o_{i,k=1}, o_{i2}, ..)$ denotes a set of outputs of a feature-class predictor $C_{FC}(i)$ at the model layer four. The input prediction vectors of the last model layer are obtained by grouping output of the fourth layer predictors as defined below

$$F_5 = ((o_{11}, o_{21}, ..)', (o_{12}, o_{22}, ..)', ..) \tag{5.4}$$

## 5.4) Inter-Slice Intra-Plane (ISAP) Model

The inter-slice intra-plane decision engine explores a VOI by analyzing slices of a plane starting at the feature-type level predictions; then feature-class level synthesization is performed followed by the plane level synthesization before reaching the final decision. Figure 5.7 illustrates ISAP structure of the modular decision engine model.

Layer 1. Feature-type Analysis: This layer is composed of feature-type modules $M_F$. The VOI is analyzed through slices considering a specific feature-type. The first prediction about each slice based on a specific feature-type is obtained at this layer. The decision engine seeks for an answer to a question like “to the what extent does the imject in the slice belong to the nodule class according to each individual feature-type?”.

Layer 2. Feature-Class Level Synthesization: This layer is composed of feature-type synthesizing modules $M_{FS}$ and feature-class modules $M_{FC}$. A feature-class level prediction about the VOI is obtained by synthesization of feature-type level results. The decision engine seeks for an answer to a question like “to the what extent does the imject in the VOI belong to the nodule class according to the geometric features?”. As an alternative, synthesization of feature-type level results can be skipped and feature-class level prediction can be applied directly; this is depicted as dashed line box ($M_{FS}$•) in Figure 5.7.

Layer 3. Plane Level Synthesization: This layer is composed of plane level modules $M_P$. The prediction about the VOI is updated considering results of feature-class level predictions. The decision engine seeks for an answer to a question like “to the what extent does the imject in the VOI belong to the nodule class according to predictions about all feature-classes for a?”.

Layer 4. Final Decision: This layer has a classifier module $M_O$ which produces the final decision about the VOI. The decision engine seeks for an answer to a question like “to the what extent does the imject in the VOI belong to the nodule class considering predictions about all?”.

## 5.5) Intra-Slice Intra-Plane (ASAP) Model

The intra-slice intra-plane decision engine explores a VOI by analyzing slices of a plane starting from the feature-type level predictions; then feature-class level synthesization is performed followed by the plane level synthesization before reaching the final decision. ASAP model differs from ISAP model such that each slice, in ASAP model, is processed individually for all feature-types and the results are synthesized to obtain an intermediate prediction while ISAP model synthesizes predictions obtained from different slices for the same feature-type. Figure 5.8 illustrates ASAP structure of the modular decision engine model.

Layer 1: Feature-type Analysis: This layer is composed of feature-type modules $M_F$. Each slice is analyzed individually considering all features. The first prediction about the slice based on all feature-types is obtained at this layer. The decision engine seeks for an answer to a question like “to the what extent does the imject in the slice belong to the nodule class according to each individual feature-?”.

Layer 2. Feature-Class Level Synthesization: This layer is composed of feature-class $M_{FC}$ and feature-class synthesizing modules $M_{FCS}$; the predictions obtained from each feature-type analysis are synthesized at this layer. The decision engine seeks for an answer to a question “to the what extent does the imject in the slice belong to the nodule class according to all feature-classes?”. Besides applying an intermediate synthesization to obtain predictions at feature-class level, the synthesization can be performed directly (at the feature-class level by $M_{FCS\bullet}$) for all feature-types from each feature-class; this is illustrated as dashed line box ($M_{FC\bullet}$) in Figure 5.8.

Layer 3. Plane Level Synthesization: This layer is composed of plane level modules $M_P$. The first prediction about the VOI is obtained based on plane level synthesization. The decision engine seeks for an answer to a question like “to the what extent does the imject in the VOI belong to the nodule class according to predictions for all slices of the plane?”.

Layer 4. Final Decision: This layer has a classifier module $M_O$ which produces the final decision about the VOI. The decision engine seeks for an answer to a question like “to the what extent does the imject in the VOI belong to the nodule class considering predictions for all planes?”.

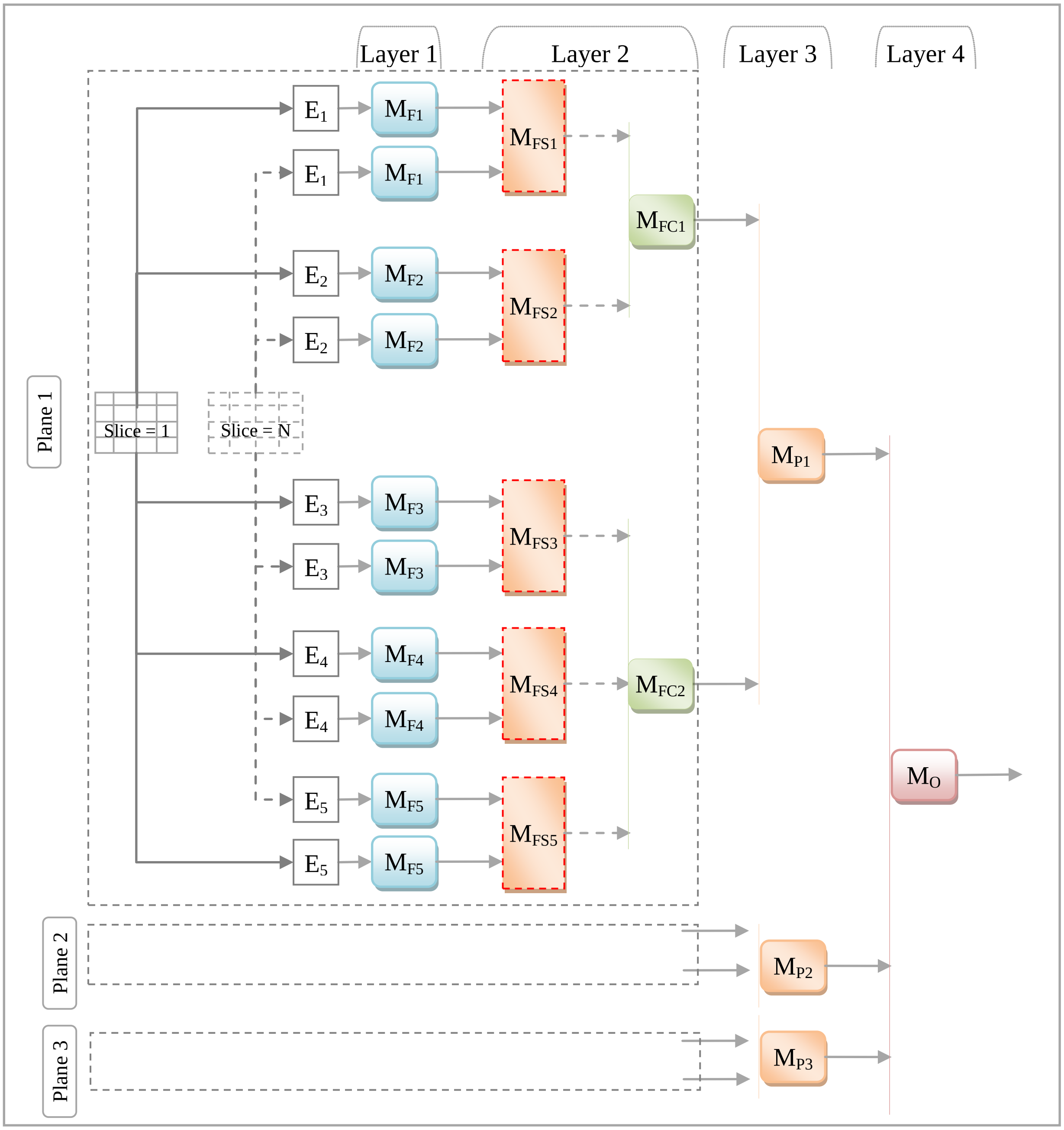


Figure 5.7 The ISAP conceptual model structure ($M_{F}$•: Feature-type level module, $M_{FS}$•: Feature-type synthesizing module, $M_{FC}$•: Feature-class level module, $M_{P•}$: Plane level module, $M_{O}$: Overall classifier module, $E_{k}$: Feature Extractor of a feature type *k*)

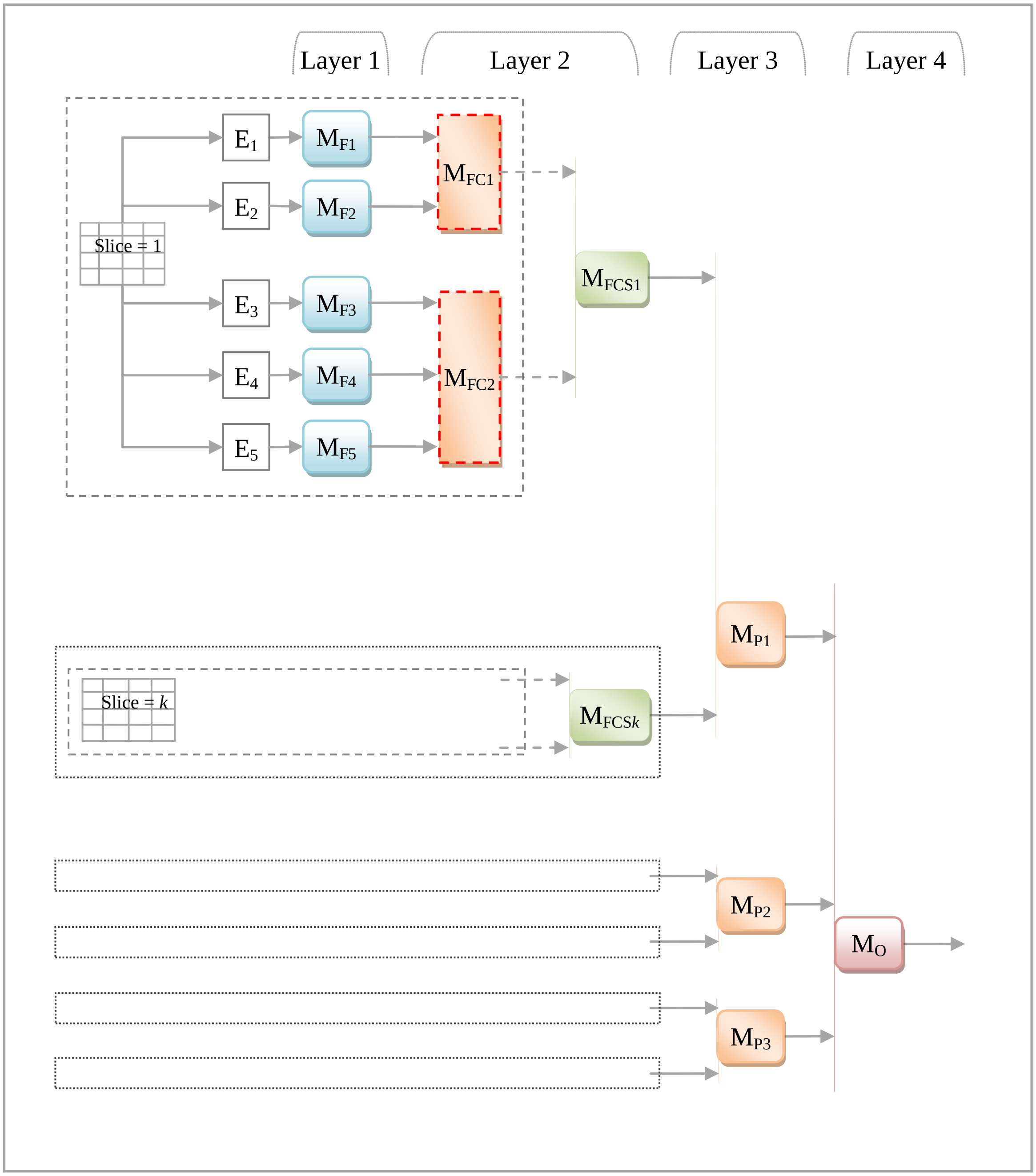


Figure 5.8 The ASAP conceptual model structure ($M_{F\bullet}$: Feature type level module, $M_{FC\bullet}$: Feature-class level module, $M_{FCS\bullet}$: Feature-class synthesis module, $M_{P\bullet}$: Plane level module, $M_O$: Overall classification module, $E_k$: Feature Extractor of a feature type $k$)

# CHAPTER 6. PERFORMANCE ASSESSMENT MEASURES

Assessment measures are used to measure goodness of the classification according to a set of classifier outputs; besides, they help to compare different classifier models in terms of tendency and error occurred during the classification process. Each assessment measure is computed over confusion matrix, which is given in Table 6.1, entries which are the number of true positives *TP*, the number of true negatives *TN*, the number of false positives *FP*, the number of false negatives *FN*, and the number of total samples *N*. The main goal of each performance measure is to produce *one* score which summarizes *four* outcomes presented in the confusion matrix of the test results. In assessment of the experiments’ result of this research, several performance measures including sensitivity, goodness of classification (GC) based on true positive rate (*GCTP*), GC based on false positive rate (*GCFP*), specificity, GC based on true negative rate (*GCTN*), GC based on false negative rate (*GCFN*), area under the ROC curve *Az*, Matthews Correlation Coefficient (*MCC*), and expected closeness value (*ECV*) are employed.

Table 6.1 Confusion Matrix

| | | CLASSIFICATION | | |
|---|---|---|---|---|
| | | Predicted as Positive (NODULE) | Predicted as Negative (NON-NODULE) | TOTAL |
| DIAGNOSIS | Positive (NODULE) | TP | FN (type II error) | P = TP + FN |
| | Negative (NON-NODULE) | FP (type I error) | TN | P’ = FP + TN |
| | TOTAL | Q = TP + FP | Q’ = FN + TN | N |

*Sensitivity* measures how many of true positive samples (nodules) out of all positive samples are detected [Kraemer]. It is defined as

$$Sensitivity = \frac{TP}{P} \tag{6.1}$$

On the other hand, sensitivity alone does not say how much agreement exists between the classifier and the diagnosis considering both true and false positives; *GCTP* can be used for this purpose. It is the number of correctly positive classified samples normalized by the average of total positive and positively predicted samples. If the *GCTP* is higher than the sensitivity, then nodules are under-predicted; if it is less, then they are over-predicted. On the other hand, it yields another comparison measure, the difference

$(FN - FP)$; the higher the difference, the higher the tendency towards type II error, i.e. missing rate of nodule will be higher; this relation is given in equation (6.3). *GCTP* is defined as

$$GCTP = \frac{TP}{(P+Q)/2} = \frac{2\,TP}{2\,TP + FN + FP} \tag{6.2}$$

$$\begin{gathered} GCTP = \frac{TP}{(P+Q)/2} > \ Sensitivity = \frac{TP}{P} \\ P = TP + FN > Q = TP + FP \\ FN - FP > 0 \end{gathered} \tag{6.3}$$

Another measure of tendency (or type I error) is *GCFP*. *GCFP* gives an idea about how much error observed in prediction of negative class samples. It is computed by first normalizing false positive predictions with average of total negative samples and total positively predicted samples; then the resulting normalized value is complemented so that it can give an idea about degree of not having type I error. It is defined as

$$GCFP = 1 - \frac{FP}{(P' + Q)/2} = \frac{TP + TN}{P' + Q} = \frac{TP + TN}{2\,FP + TP + TN} \tag{6.4}$$

*GCFN* is employed in analysis of type II error. *GCFN* is a measure of how many positive class samples is missed by the classifier; *GCFN* is obtained by normalizing the number of false negative predictions with the average of the total positive samples and the total negatively predicted samples; then the resulting normalized value is complemented so that it can give an idea about degree of not having type II error. In case of nodule detection, *GCFN* score is crucial due to cost of life. It is defined as

$$GCFN = 1 - \frac{FN}{(P + Q')/2} = \frac{TP + TN}{P + Q'} = \frac{TP + TN}{2\,FN + TP + TN} \tag{6.5}$$

Similar to sensitivity, *specificity* measures how many of true negative samples (non-nodules) out of all negative samples are detected [Kraemer]. It is defined as

$$Specificity = \frac{TN}{P'} \tag{6.6}$$

Again, specificity alone does not say how much agreement exists between the classifier and the diagnosis considering both true and false negative classes; *GCTN* can serve for this purpose. If the *GCTN* is higher than the specificity, then an under-prediction of non-nodules is observed; if it is less, then over-prediction exists. The higher the difference $(FP - FN)$ is, the higher the type I error exists; this relation is given in equation (6.8). *GCTN* is defined as

$$GCTN = \frac{TN}{(P' + Q')/2} \tag{6.7}$$

$$GCTN = \frac{TN}{(P' + Q')/2} > \ Specificity = \frac{TN}{P'}$$
$$P' = TN + FP > Q' = TN + FN \tag{6.8}$$
$$FP - FN > 0$$

The area under the ROC curve *Az* is another measure of classification performance in terms of sensitivity and complement of specificity [Lasko05]. The ROC curve is obtained by applying some threshold values to the classifier output and then calculating corresponding true positive rates (*sensitivity*) and false positive rates (1 - *specificity*). The max value of the area is 1 which refers to the perfect classification and the min value is 0.5 which refers to the poorest one; i.e. the classifier yields a result no better than tossing a coin. The area can be interpreted in different ways: a) average sensitivity b) average specificity c) the probability that a test will produce an output which is higher for a randomly selected diseased subject than for a randomly selected healthy one [Metz86, Obuch03, Lasko05].

The correlation coefficient which is also known as Matthews correlation coefficient (*MCC*) in the literature, [Baldi] measures the correlation between diagnosis and classification outputs. Given target set $T = \{t_i\}$ and classification output sets $O = \{o_i\}$, means $\bar{t}$ and $\bar{o}$, and standard deviations $\sigma_T$ and $\sigma_O$ for target and classification sets, respectively, *MCC* is defined by [Baldi] as

$$MCC = \sum_i \frac{(t_i - \bar{t})}{\sigma_T} \frac{(o_i - \bar{o})}{\sigma_O} \tag{6.9}$$

It gives an index of how targets and corresponding classification outputs are aligned in sign and value using their normalized values by their sample mean and standard deviations. *MCC* can be further expressed in terms of confusion matrix elements for two classes with two categories as

$$MCC = \frac{TP\ \times TN - FP\ \times FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}} \tag{6.10}$$

*ECV* performance score measures complement of average error at the output of the classifier compared to the corresponding target value. It is defined as

$$ECV = 1 - \frac{\sum_i |t_i - o_i|}{N} \tag{6.11}$$

It gives an idea how close output values to the target before applying a threshold which is utilized to decide the class of the input feature.

# CHAPTER 7. ANALYSIS OF RESULTS

In this research, we used 80 nodules and non-nodules imjects in total with 40 imjects from each class. We evaluated our detection technique using a test data set that has 16 imject samples of 8 nodules and 8 non-nodules. The training data set has 48 imjects and validation data set has 16 imjects; the both data set has equal number of nodule and non-nodule samples. We tested our model employing six experiment groups; each group has several sub-groups. In the following sections, we analyze results of each experiment group.

## 7.1) Experiment Group A

In this experiment group, we applied the Levenberg-Marquet learning algorithm using batch processing approach. We conducted 72 experiments with changing training parameters mu_dec = {0.05, 0.1, 0.5}, mu_inc = {5, 10}, mu_max = {100, 1000, 10e10}, epochs = {10, 10, 500, 1000}, and keeping others constant which are mu = 0.001, time = 60 min max, goal = 0, min_grad = 1e-10, max_fail = 3. Significant of these parameters are explained in the Section (2.5). The network structure is composed of 1 hidden layer. The number of nodes in the hidden layer of each predictor is given in Table 7.1; the output layer of each ANN has 1 node.

Table 7.1 Number of hidden nodes in each ANN predictor (ML: Model Layer)

| Feature Class | Feature Type | ML 1 | ML 2 | ML 3 | ML 4 | ML 5 |
|---|---|---|---|---|---|---|
| Geometric | Spectral | 2 | 1 | 2 | 2 | 2 |
| | Circularity | 9 | 1 | 2 | | |
| Photometric | Co-occurrence | 11 | 1 | 2 | 2 | |
| | Run-length | 12 | 1 | 2 | | |

### 7.1.1) Comparison of Experimental Parameters

In this section we analyze nodule and non-nodule detection performance based on assessment measures introduced in Chapter 6. Also, we extend our analysis by exploring tendency towards each imject type (type I and II errors), accuracy of Decision Engine output, correlation between classification results and actual classes, variation in performance scores, and all parameter sets observed to find a set which yields better classification performance. Figure 7.1 depicts a summary of classification performances at the last layer of the Decision Engine for the experiments group-A; in subsections, all analysis refers to Figure 7.1 otherwise said.

#### 7.1.1.1) Goodness of Classification for Nodule Imjects

Max-curve reads that the Decision Engine classifies all nodules perfectly for some parameter sets (with Sensitivity = 100%). We can infer from the median and mean curves that most of the parameter sets yield a classification performance above the mean according to sensitivity and *GCTP* medians which are 88% and 82%, respectively. This indicates that, in classification of nodule imjects, the Decision Engine is not affected too much by the parameters selected for a sample set. The MIN curve shows that some of the parameter sets cannot yield a good classification.

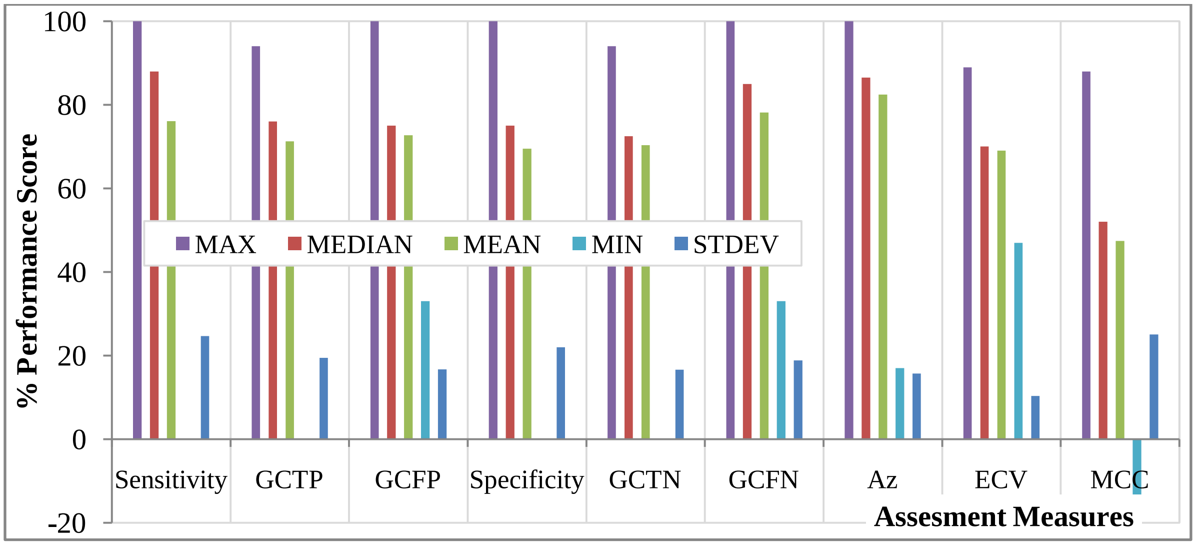


Figure 7.1 Overall classification performance of the Decision Engine against parameter variations (summary of all experiments for the final layer)

### 7.1.1.2) Tendency to Nodule (type I error analysis)

The max *GCFP* value among the experiments is 100%; that is the Decision Engine does not have a prediction tendency towards nodules for some parameter sets. Since the median difference between Sensitivity and *GCTP* is 6%, then we can conclude that there is a slight tendency towards the nodule prediction in most of the parameter sets. We can have a similar conclusion when we compare medians for sensitivity = 88% against specificity = 75% and *GCTP* = 82% against *GCTN* = 80%. Also *GCFN* = 87% is larger than *GCFP* = 76%, that is type II error is less than type I error in most of the experiments, this is good for a nodule detector.

### 7.1.1.3) Goodness of Classification for Non-nodule Imjects

According to the max-curve, the Decision Engine classifies all nodules perfectly for some parameter sets (with specificity = 100%). The median and mean curves read that most of the parameter sets yield a classification performance above the mean according to specificity (75%) and *GCTN* (80%). This indicates that, in classification of non-nodule imjects, the Decision Engine is not affected too much by the parameters observed.

### 7.1.1.4) Tendency to Non-nodule Imjects (type II error analysis)

The max *GCFN* value is 100%; that is the Decision Engine does not have a tendency towards non-nodules for some parameter sets. The median value of *GCFN* (87%) is slightly above the mean value (85%) and below the max value (100%); i.e. more than half of the parameter sets does not produce a result in favor of non-nodule prediction which is a parallel result with type I error analysis.

### 7.1.1.5) Accuracy of the Classification Outputs

Some parameter sets produce a perfect prediction values (*ECV* = 100%). The accuracy (*ECV*) of the Decision Engine is 73% on average. Most of the Decision Engine output values (median of 74%) are above the mean.

### 7.1.1.6) Correlation between Prediction and Actual Classes

The max correlation between predictions and true classes is 100% as seen with *ECV*. The median *MCC* value 63% depicts that most of the experiment sets produce predictions which are correlated with the true class of imjects. Notice that *MCC* is a measure of goodness of classification considering prediction results of both classes.

### 7.1.1.7) Analysis of Performance Variations

Variation in performance is 30%, 28%, 28%, and 27% among specificity, among *MCC*, and among *GCTN* performance scores, respectively. This means that they are more sensitive than other measures to the parameter changes. The variation is 16% for *GCTP*, that is the nodule detection by the Decision Engine are affected less than non-nodule detection. The variation is 15% for *GCFN* and 20% for *GCFP*; we can conclude that type II error is less sensitive to the parameter differences than type I error. The variation among *ECV* scores is 14%, which is the min among the others; that is prediction output values are less sensitive to parameter variation.

### 7.1.1.8) The Parameter Set Which Yields a Better Classification Performance

Parameter sets of some selected experiments are shown in Table 7.2 and their corresponding performance scores is given in Table 7.3. The correlation among the experiment parameters are shown in Table 7.4. As seen from Table 7.4, the elapsed time is not sensitive to these parameter variations. Table 7.5 to Table 7.12 show the confusion matrixes obtained from the experiment 42 for each Decision Engine layers.

Table 7.2 Parameters of some selected experiments from group-A

| | Epochs | mu_dec | mu_inc | mu_max | Elapsed Time (sec.) |
|---|---|---|---|---|---|
| Exp-42 | 500 | 0.05 | 10 | 1E+11 | 19 |
| Exp-26 | 50 | 0.1 | 5 | 1000 | 20 |
| Exp-33 | 50 | 0.5 | 5 | 1E+11 | 11 |
| Exp-55 | 1000 | 0.05 | 5 | 100 | 13 |
| Exp-69 | 1000 | 0.5 | 5 | 1E+11 | 12 |

Table 7.3 Goodness of classification scores for experiment 42 in group-A

| Sensitivity | GCTP | GCFP | Specificity | GCTN | GCFN | Az | ECV | MCC |
|---|---|---|---|---|---|---|---|---|
| 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |

Table 7.4 Correlation between elapsed time and group-A experimental parameters

| | mu_dec | mu_inc | mu_max |
|---|---|---|---|
| Elapsed time | -0.16 | -0.01 | 0.01 |

Table 7.5 Confusion matrix of model layer 1 (Exp-42)

| | Spectral Feature | Circularity | CoOccurence | RunLength |
|---|---|---|---|---|
| True Positive | 711 | 925 | 988 | 866 |
| False Negative | 489 | 275 | 212 | 334 |
| False Positive | 412 | 324 | 192 | 675 |
| True Negative | 788 | 876 | 1008 | 525 |

Table 7.6 % Confusion matrix of model layer 1 (Exp-42)

| | Spectral Feature | Circularity | CoOccurence | RunLength |
|---|---|---|---|---|
| True Positive | 30 | 39 | 41 | 36 |
| False Negative | 20 | 11 | 9 | 14 |
| False Positive | 17 | 14 | 8 | 28 |
| True Negative | 33 | 37 | 42 | 22 |

Table 7.7 Confusion matrix of model layer 2 (Exp-42)

| | Spectral Feature | Circularity | CoOccurence | RunLength |
|---|---|---|---|---|
| True Positive | 24 | 24 | 22 | 15 |
| False Negative | 0 | 0 | 2 | 9 |
| False Positive | 16 | 11 | 4 | 10 |
| True Negative | 8 | 13 | 20 | 14 |

Table 7.8 % Confusion matrix of model layer 2 (Exp-42)

| | Spectral Feature | Circularity | CoOccurence | RunLength |
|---|---|---|---|---|
| True Positive | 50 | 50 | 46 | 31 |
| False Negative | 0 | 0 | 4 | 19 |
| False Positive | 33 | 23 | 8 | 21 |
| True Negative | 17 | 27 | 42 | 29 |

Table 7.9 Confusion matrix of model layer 3 (Exp-42)

| | Spectral Feature | Circularity | CoOccurence | RunLength |
|---|---|---|---|---|
| True Positive | 8 | 8 | 8 | 6 |
| False Negative | 0 | 0 | 0 | 2 |
| False Positive | 4 | 2 | 0 | 5 |
| True Negative | 4 | 6 | 8 | 3 |

Table 7.10 % Confusion matrix of model layer 3 (Exp-42)

| | Spectral Feature | Circularity | CoOccurence | RunLength |
|---|---|---|---|---|
| True Positive | 50 | 50 | 50 | 38 |
| False Negative | 0 | 0 | 0 | 13 |
| False Positive | 25 | 13 | 0 | 31 |
| True Negative | 25 | 38 | 50 | 19 |

Table 7.11 Confusion matrix of model layer 4 and 5 (Exp-42)

| | Geometric | Photometric | Overall |
|---|---|---|---|
| True Positive | 8 | 8 | 8 |
| False Negative | 0 | 0 | 0 |
| False Positive | 4 | 0 | 0 |
| True Negative | 4 | 8 | 8 |

Table 7.12 % Confusion matrix of model layer 4 and 5 (Exp-42)

| | Geometric | Photometric | Overall |
|---|---|---|---|
| True Positive | 50 | 50 | 50 |
| False Negative | 0 | 0 | 0 |
| False Positive | 25 | 0 | 0 |
| True Negative | 25 | 50 | 50 |

## 7.1.2) Change of Classification Performance through Each Decision Engine Layer

In this section we analyze prediction performance for experiment 42 (number of epochs = 500, mu_dec = 0.05, mu_inc = 10, mu_max = $10^{10}$, and elapsed time = 19 sec). As seen from Figure 7.2, the classification performance improves from the first layer of the Decision Engine to the final layer, in general.

### 7.1.2.1) Nodule Detection Performance

At the first layer of the Decision Engine, Figure 7.2 shows the overall nodule detection scores which are 73% (sensitivity), 71% (*GCTP*), 79% (Az), 68% (*GCFP*) and 72% (*GCFN*), and 65% (*ECV*). These results, and also the comparison of GCTP versus sensitivity, show that nodule detection rate is slightly higher than the non-nodule detection while the type II error rate is slightly lower than the type I error; accuracy of predictions (*ECV*) is somehow above the toss-a-coin (50%) level. Figure 7.3 portrays performance measures of each predictor through model layers; the nodule detection performance by the spectral feature is 59%, 61%, 65% for sensitivity, *GCTP*, *GCFP*, respectively, while they are 77%, 76%, and 74% if the circularity shape feature is employed; when co-occurrence feature is used, the performance measures increases to 82%, 83%, 84% and run-length feature yields around 72%, 63%, and 51% as shown in Figure 7.3. Among the feature type predictors, the co-occurrence based prediction gives the highest score, then circularity based prediction follows; the min nodule detection result is obtained when the spectral based prediction employed, but it yields less type I error than the run-length based prediction. The *Az* score depicts the similar results. Predictors other than the spectral-based one show higher performance for type II error as compared to type I error. The difference (*GCFN – GCFP*) is the highest for run-length predictor (17%); it is -4%, 3%, and -2% for the spectral, circularity, and co-occurrence based predictions, respectively. When we analyze the results for each plane using the Figure 7.4, we can see that prediction results are not the same for each plane slices.

The second layer modules, on which the slice level analysis is applied by synthesizing prediction results of the same plane slices, improve the nodule detection rate as Figure 7.2 depicts; sensitivity (89%) and type II (86%) scores increase while type I error (63%) decreases. As seen from Figure 7.3, the both geometric feature-based predictors reach the perfect performance as regard to the sensitivity and type II error. On the other hand, the type I error performance gets worse for geometric based predictions. That means this class of predictors over-predict nodule samples. This can be also concluded from the tables Table 7.5 to Table 7.8 where *TP* = 30 (50), *FP* = 17 (33) in the first layer (second layer). The difference (*GCFN – GCFP*) is 50%, 37%, 7%, 3% for the spectral, circularity, co-occurrence, and run-length based predictors, respectively. This shows that type I error increases versus type II error for the geometric predictors. The reason for such a result may be due to using ‘cube normalization’ as discussed in the Section (4.3) (A Discussion Related to Invariance Issue). Analysis of results for slices of a plane shows that the geometric based prediction produces almost the same result while photometric based prediction performance keeps changing among planes as seen in the Figure 7.4.

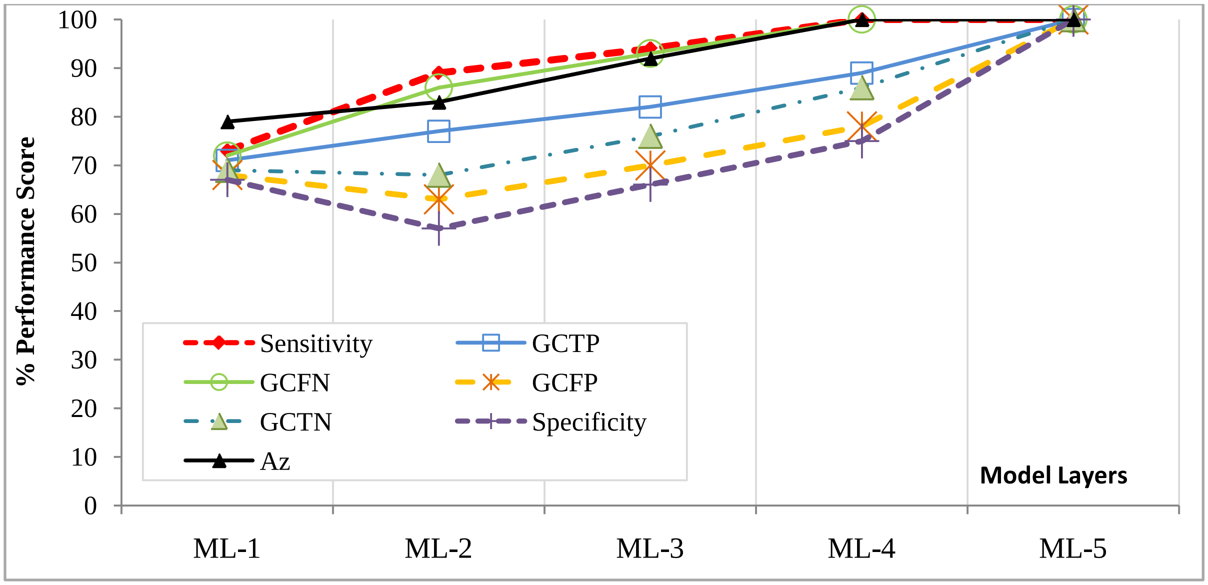


Figure 7.2 Average change of classification performance through Decision Engine Layers

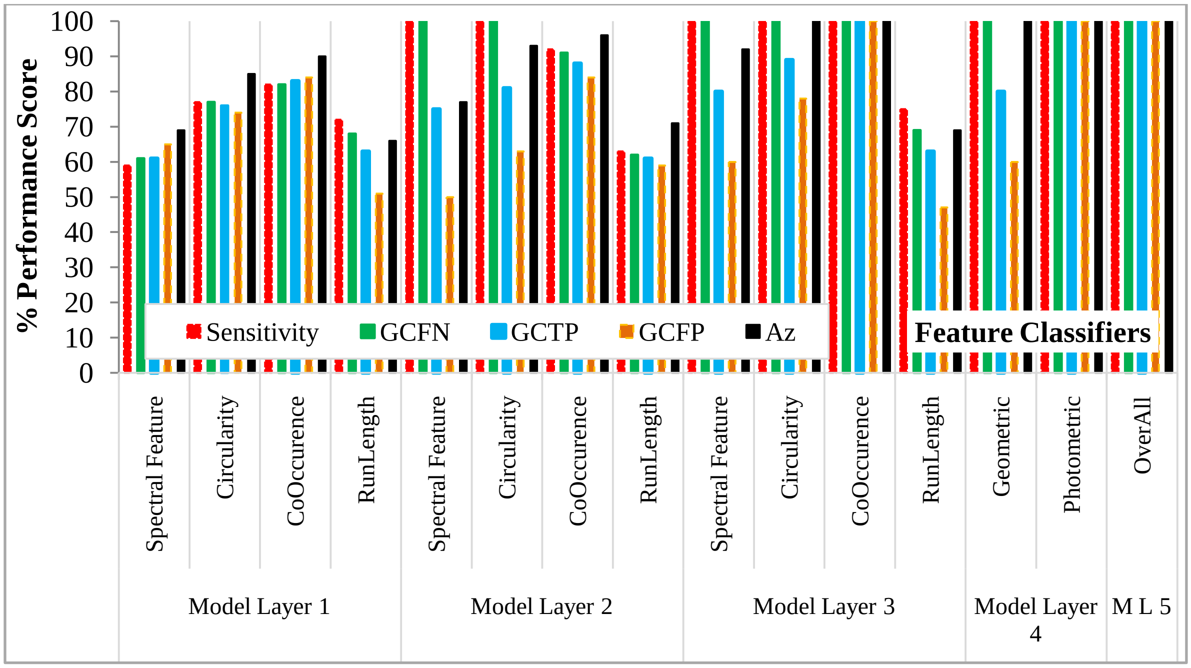


Figure 7.3 Change of classification performance in the Decision Engine layers (Analysis of nodule detection)

The third layer that combines decisions from all planes improves the decision made in the previous layers as Figure 7.2 depicts; sensitivity (94%), type II (93%), and type I error (70%) scores increase from 89%, 86%, and 63% respectively. This shows that non-nodule detection rate also improves at this model layer. Figure 7.3 portrays a detail look for each individual feature predictor; geometric feature-based prediction reaches the perfect score for nodule detection and type II error while still showing some type I errors (*GCFP* = 60% and 78%). On the other hand, the co-occurrence-based prediction reaches the perfect score at each performance score while the run-length based prediction has a slight improvement.

At the fourth layer, both feature class-based predictions produce the perfect performance for nodule detection and type II error, but the geometric feature prediction module still yields type I error (GCFP 60%). The Engine reaches the perfect classification at the final layer.

### 7.1.2.2) Non-Nodule Detection Performance

At the first layer of the Decision Engine, Figure 7.2 shows the overall nodule detection scores which are 67% (specificity), 69% (*GCTN*); as we mentioned earlier non-nodule detection rate is slightly less than nodule detection rate. The spectral (circularity) based predictor scores are specificity of 66% (73%) and *GCTN* of 64% (75%) while the co-occurrence based predictor scores are specificity of 84% (44%) and *GCTN* of 83% (51%) as seen in Figure 7.5. The co-occurrence based prediction is the highest as it was for nodule prediction at layer 1. The Figure 7.6 depicts prediction result for slices of each plane. The performance differs from plane to plane for each feature type; for example, the geometric feature based predictors yield a higher score in slices of plane 2 while the co-occurrence based prediction is high on slice of the first plane and run-length predictor is good on the slices of the plane 3 according to specificity measure.

For the second layer modules, the specificity score decreases from 67% to 57% in overall predictions as seen in Figure 7.2. This decline may arise from the geometric based predictions; as seen in Figure 7.5, both geometric predictions have a lower specificity than the first layer's predictions of this feature class.

In the third layer, overall prediction improves as seen in Figure 7.2. The co-occurrence prediction has the highest score and circularity, spectral, and run-lengths follow in the order. At the fourth layer, specificity keeps increasing and it reaches the perfect score at the final layer.

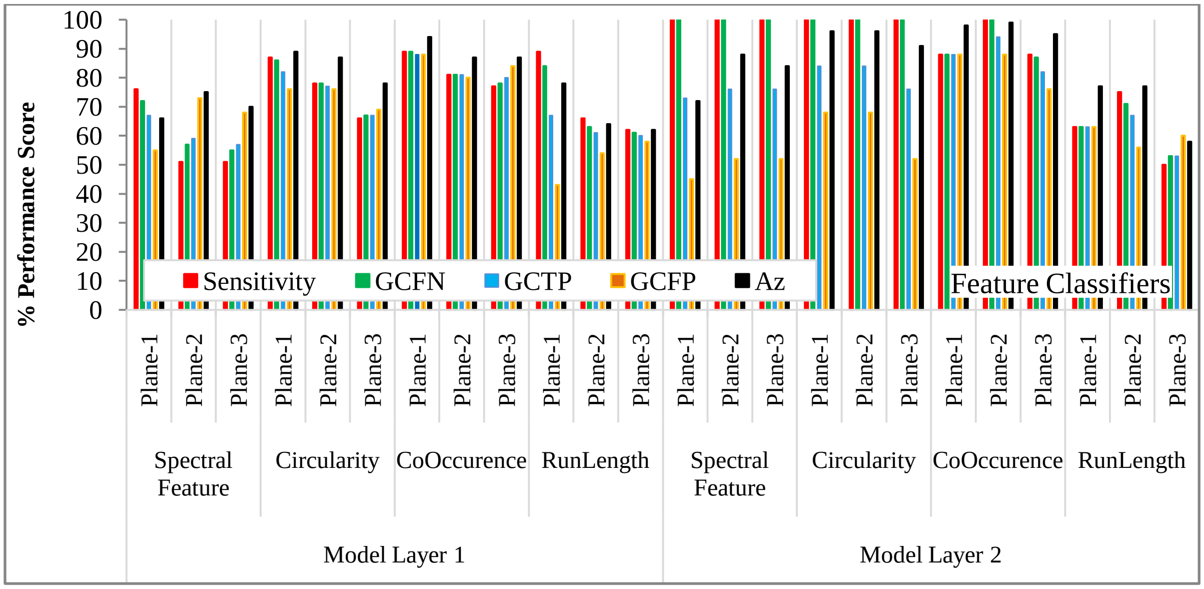


Figure 7.4 Change of classification performance in the Decision Engine layers 1 and 2 (Analysis of nodule detection at slice level for each plane)

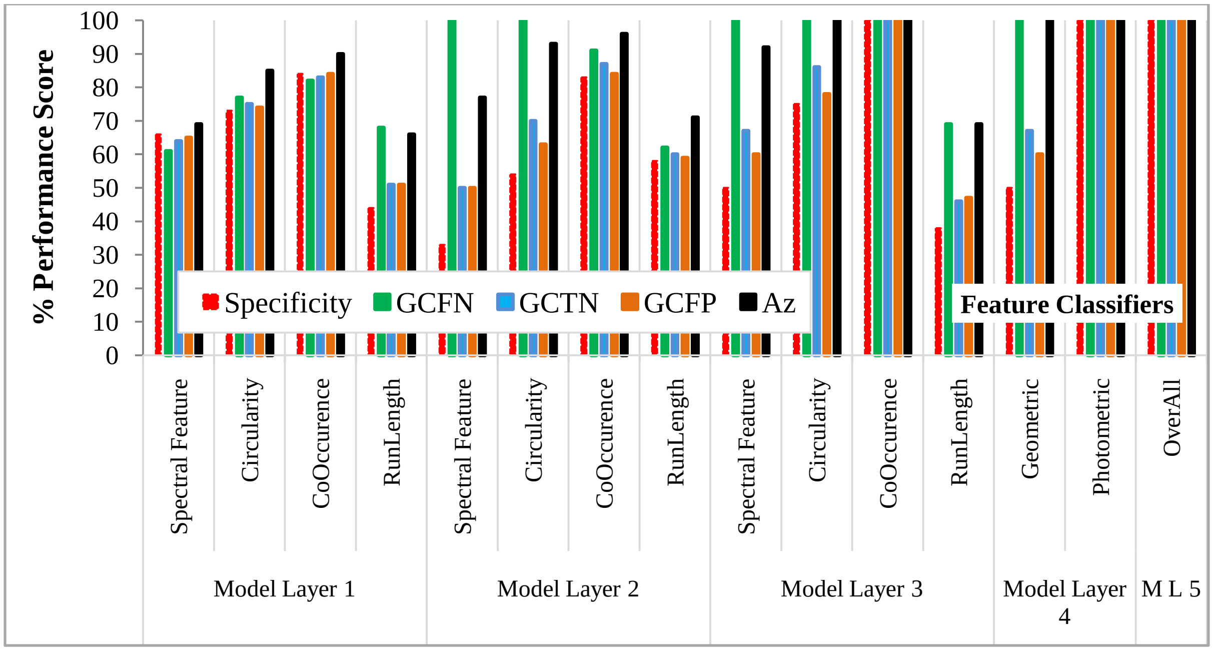


Figure 7.5 Change of classification performance in the Decision Engine layers (Analysis of non-nodule detection)

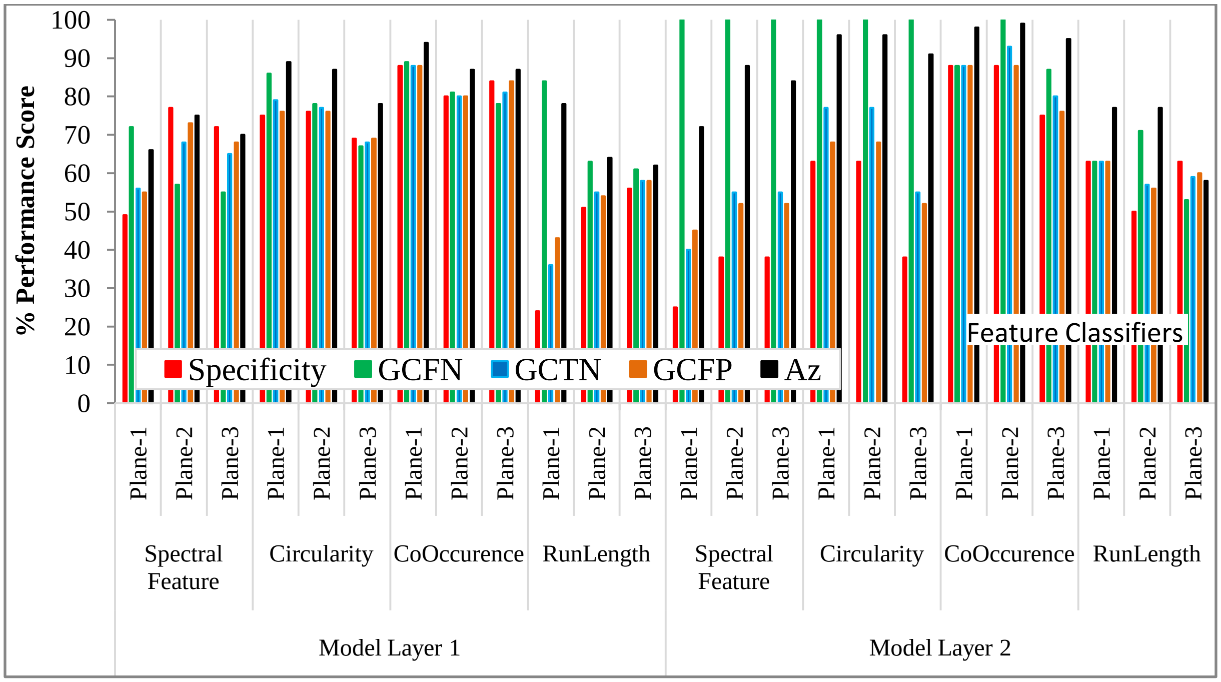


Figure 7.6 Change of classification performance in the Decision Engine layers 1 and 2(Analysis of non-nodule detection at slice level for each plane)

## 7.2) Experiment Group B

In this experiment group, we applied the same parameters which are used in group-A; the only difference is that all ANNs have only one node at the hidden layers. The comparison of Figure 7.1 and Figure 7.7 show that the group one parameter sets produce slightly better result than the second group overall. The experiment 62 of this group produced the best detection result; the parameters of this experiment are: 1) epoch = 1000, 2) mu_dec = 0.1, 3) mu_inc = 5, 4) mu_max = 1000, and 5) max_fail =

3. Figure 7.8 depicts the detection performance of the experiment 62; even sensitivity and type II error (*GCFN*) performances reach the perfect score at the final layer, other scores did not reach that level. Therefore, some non-nodule imjects could not be detected by the engine. The performance of group-A at the model layer 1 is slightly higher than the performance of the group-B; we can conclude from this result that increasing the number of hidden nodes may not yield a big performance difference.

The parameters for some selected experiments from group-B are listed in the Table 7.13. The comparison of the Table 7.2 with the Table 7.13 shows that ANN structures with different number of hidden nodes may produce similar result using different parameters.

Table 7.13 Parameters of some selected experiments from group-B

| Experiment Number | Epochs | mu_dec | mu_inc | mu_max | Elapsed Time (sec.) |
|---|---|---|---|---|---|
| Exp-62 | 1000 | 0.1 | 5 | 1000 | 8 |
| Exp-19 | 50 | 0.05 | 5 | 100 | 8 |
| Exp-60 | 1000 | 0.5 | 10 | 1010 | 10 |
| Exp-59 | 1000 | 0.05 | 10 | 1000 | 8 |
| Exp-63 | 1000 | 0.1 | 5 | 1010 | 8 |

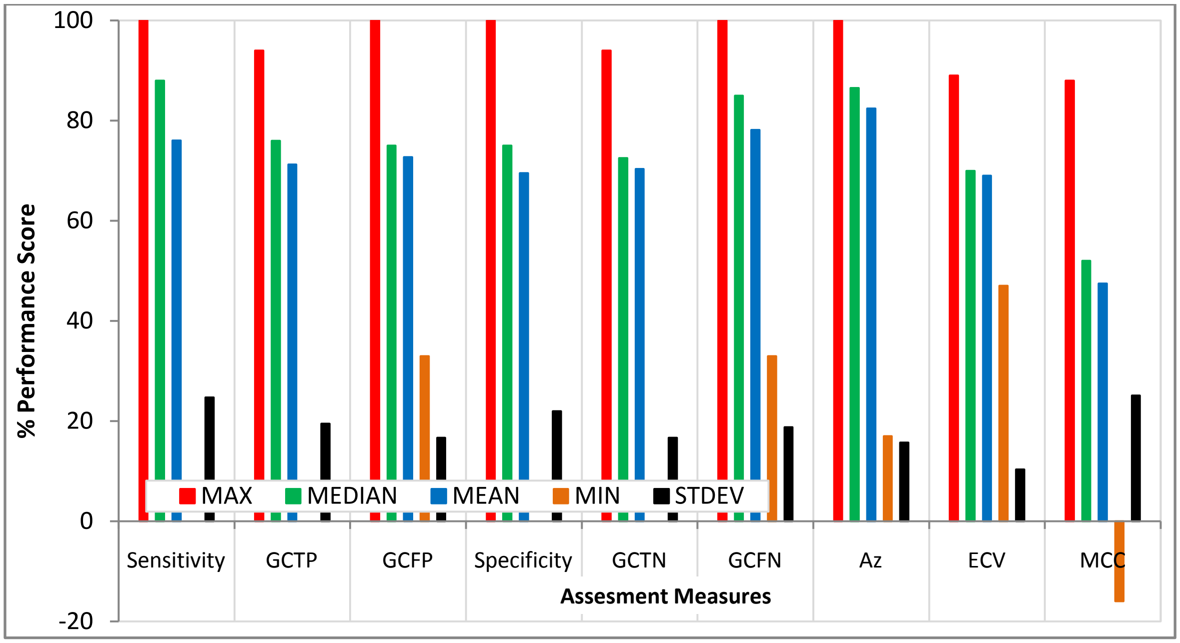


Figure 7.7 Overall classification performance of the Decision Engine against parameter variations (summary of all group-B experiments)

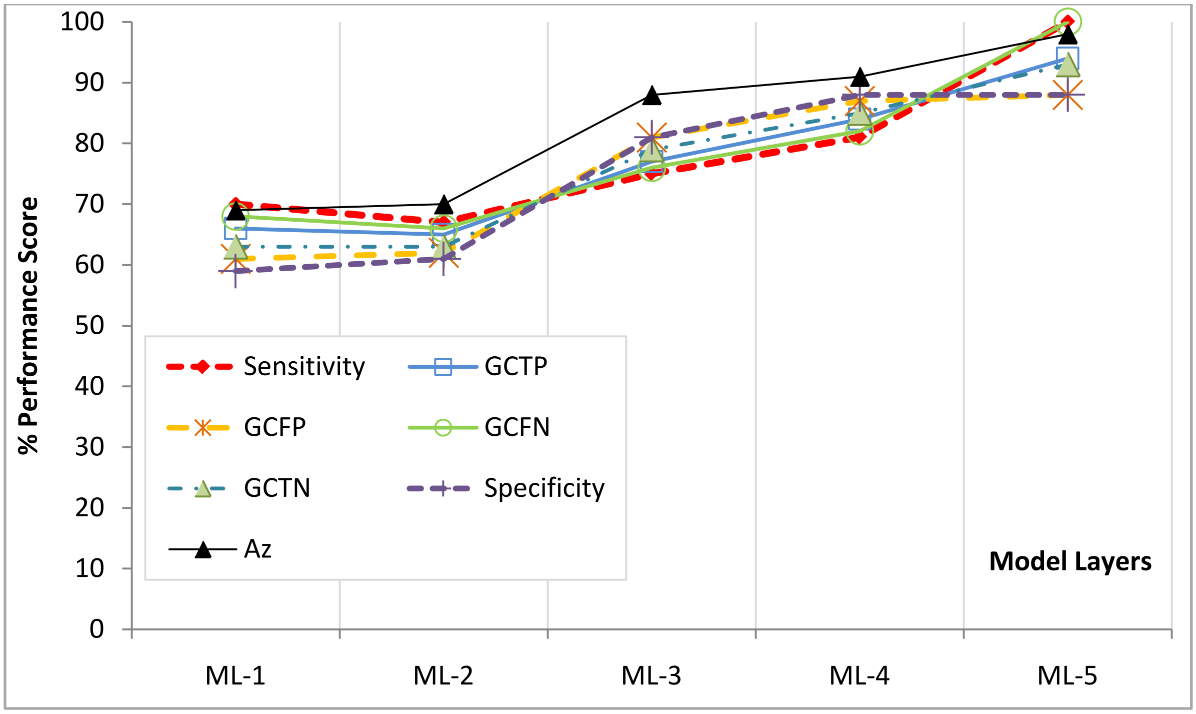


Figure 7.8 Average change of classification performance through Decision Engine layers for experiments group-B

## 7.3) Experiment Group C

Experiment group-C is conducted to explore effect of increasing training time by allowing the network to continue training until one of the learning performance value is achieved; i.e. making the parameter max_fail as 3 (experiment G3) and infinity (experiment G3_1). In this experiment group, all the parameters are the same with the group-B except max_fail = Inf and the number of hidden layer nodes at the model layer 1 is the same as in the group-A, see the Table 7.1. The Table 7.14 shows the parameter set for the experiment which produced the best result. The elapsed time difference is not so high but the detection rate is significantly higher at the model layer 1 and 4, even both models reach the perfect score at the final layer. At the model layer 3, specificity and type 1 error decline when G3 set is used; this would arise due to lack of time allowed for adjusting weights of the ANNs at this layer. So the network could not be trained for a better generalization.

Table 7.14 Parameters of the selected experiments from group-C

| | epochs | mu_dec | mu_inc | mu_max | max_fail | elapsed time (sec.) |
|---|---|---|---|---|---|---|
| G3 (Exp-18) | 10 | 0.5 | 10 | 10^10 | 3 | 10 |
| G3_1 (Exp-33) | 50 | 0.5 | 5 | 10^10 | Inf | 36 |

The Figure 7.9 and Figure 7.10 depict overall classification performance of the Decision Engine used in G3 and G3_1 experiments, respectively. As seen from the figures, performance of the engine is not affected so much for more than half of the parameter settings; besides, G3_1 responds better than G3 against parameter variations.

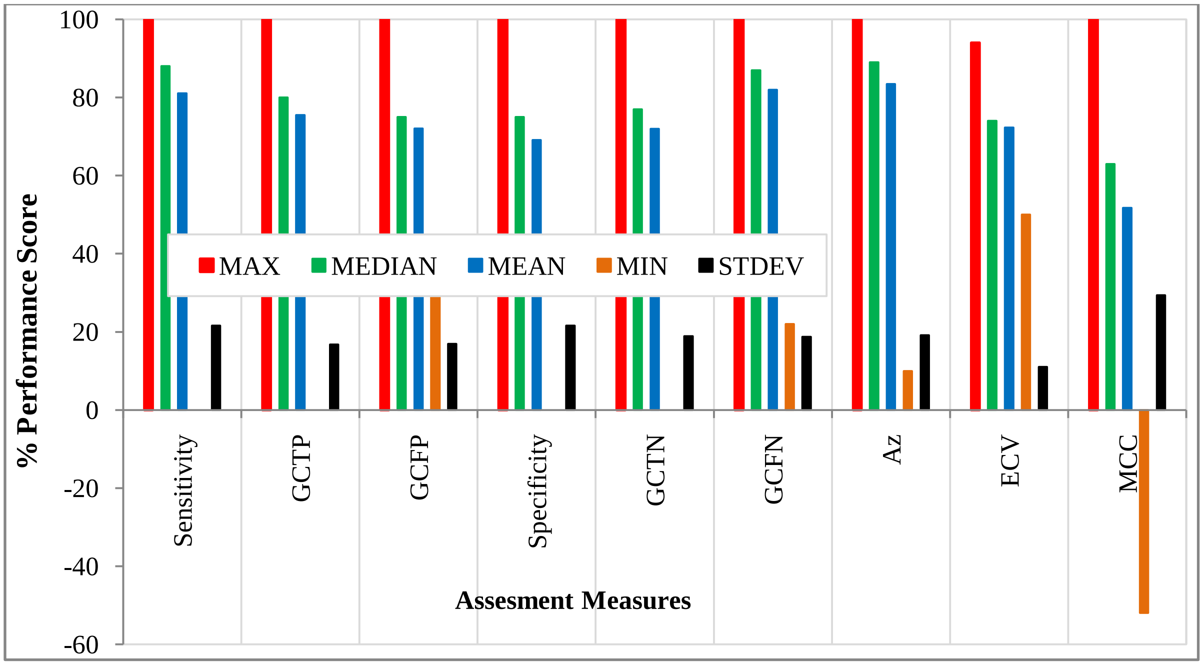


Figure 7.9 Overall classification performance of the Decision Engine against parameter variations (summary of all G3 experiments)

A closer look at each predictor type is shown in the Figure 7.10 which is a detailed version of the Figure 7.9 for the model layer 1. As the figure depicts, G3_1 set produces a better result, in general; on the other hand, the spectral feature classifier performance of G3_1 set is lower than the performance of G3 set at the plane 1 according to specificity and type 1 error measures. This may arise from an over-generalization for the non-nodule samples.

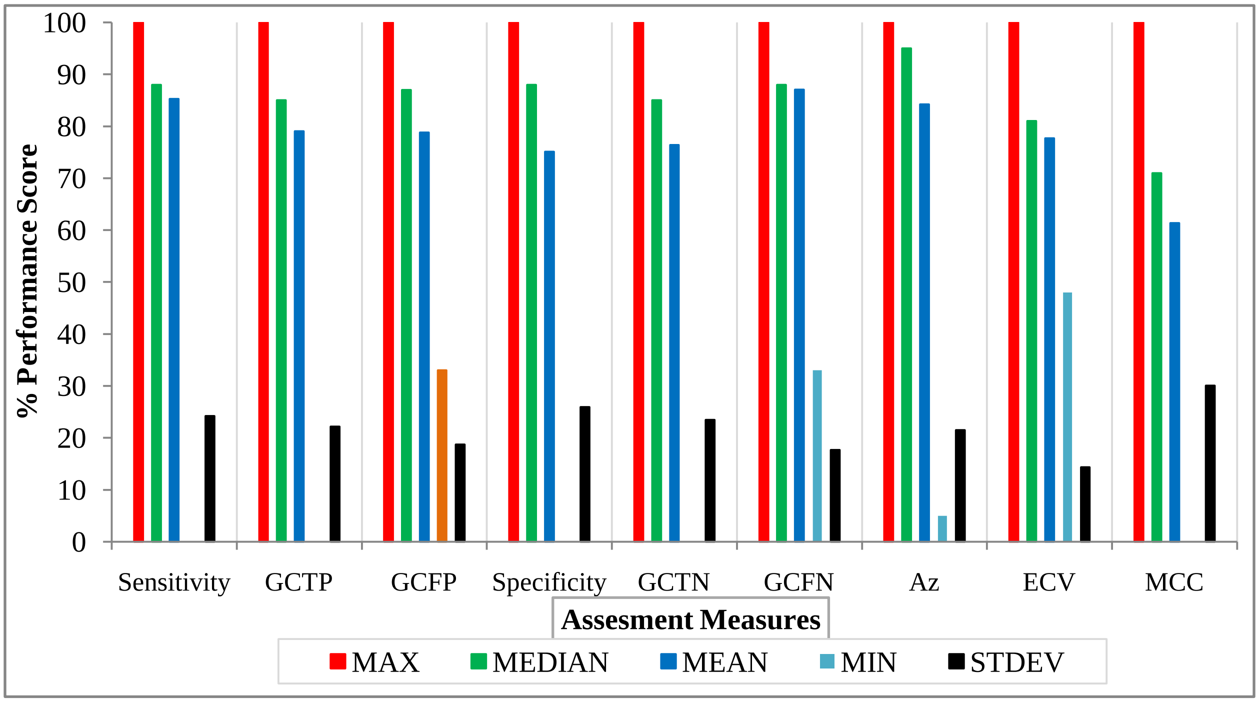


Figure 7.10 Overall classification performance of the Decision Engine against parameter variations (summary of all G3_1 experiments)

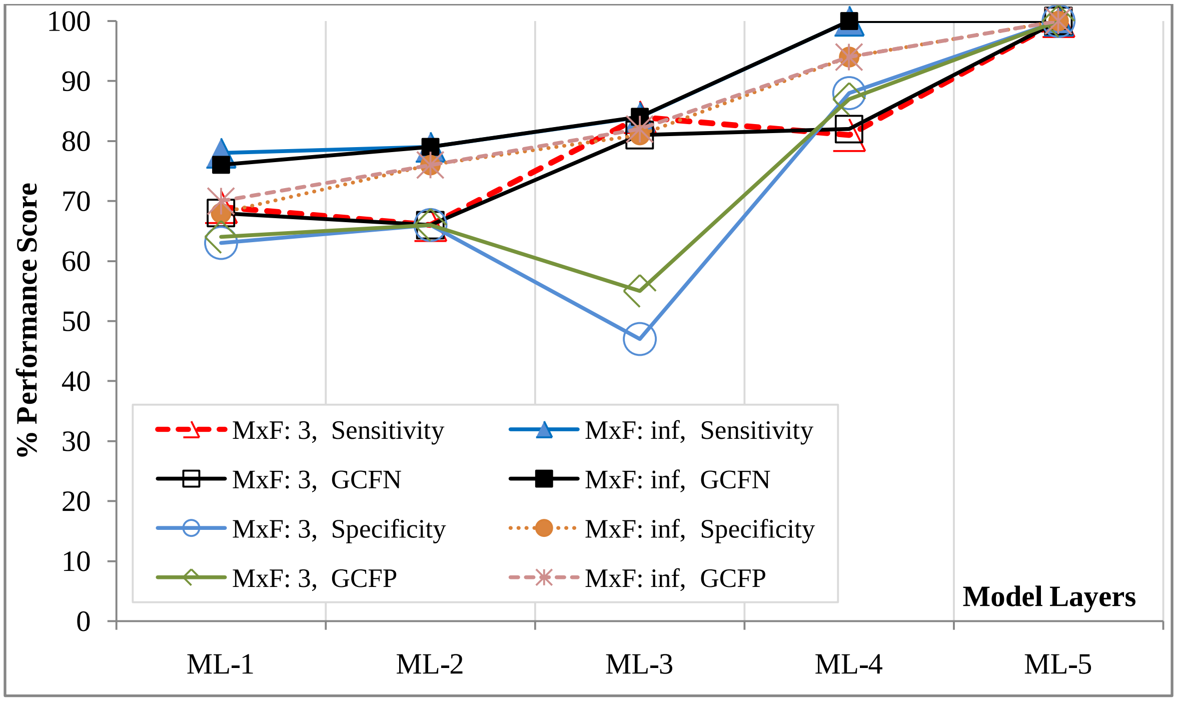


Figure 7.11 Comparison of Engine Models with different taining times (MxF: Max_fail)

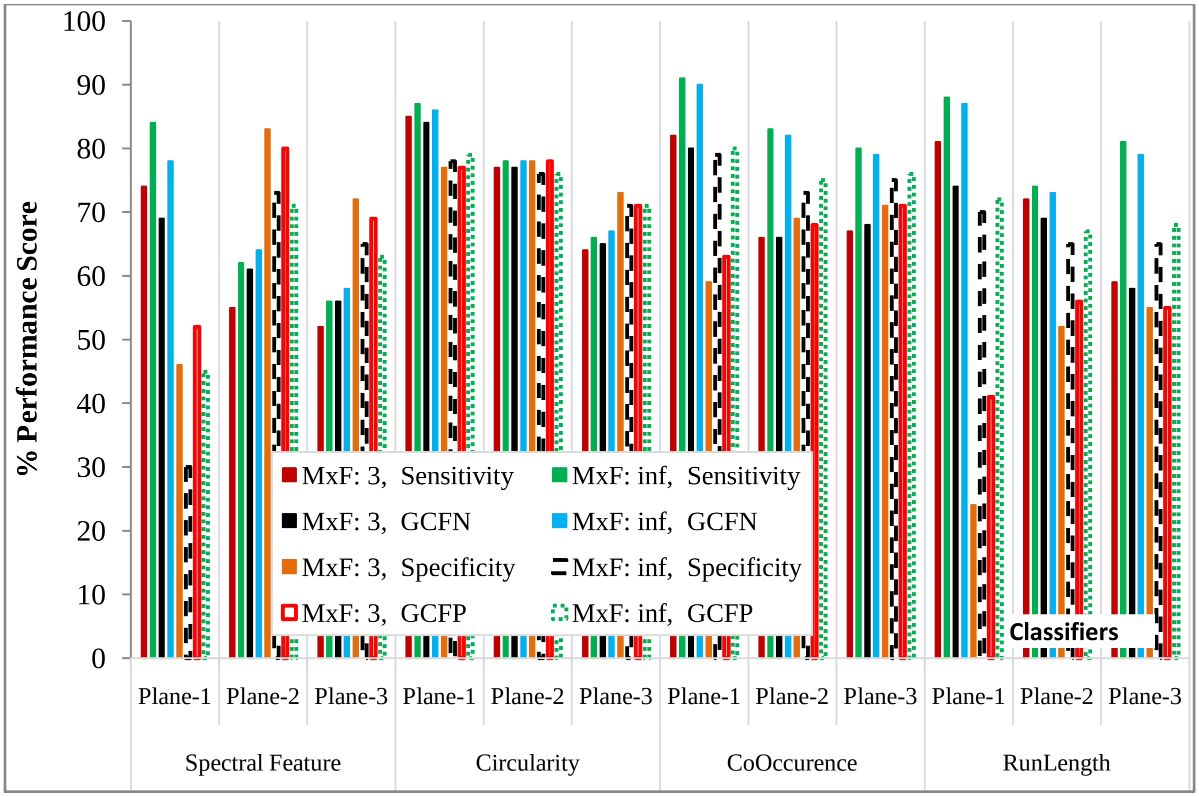


Figure 7.12 Comparison of Engine Models with different taining times (Model layer 1 detail analysis, MxF: Max_fail)

## 7.4) Experiment Group D

In this experiment group, we explored effect of adding extra network layer while keeping the number of hidden nodes the same; hence this group is another version of the experiment G3_1 with some extra hidden layers. We experimented additional two sets of ANN structures, namely G5 and G6, where G5 has two hidden layers while G6 has three hidden layers at the first model layer of the Decision Engine. The number of hidden nodes is given in Table 7.15. The parameter sets are given in Table 7.16. The execution time for the network structure of 3 hidden layers was 130 sec while it was 432 sec when two hidden layers were used.

Table 7.15 Number of nodes in hidden layers for experiment group-D (hl: Hidden layer)

| Feature Class | Feature Type | G5 | | G6 | | |
|---|---|---|---|---|---|---|
| | | hl-1 | hl-2 | hl-1 | hl-2 | hl-3 |
| Geometric | Spectral | 1 | 1 | 1 | 1 | n/e |
| | Circularity | 5 | 4 | 6 | 2 | 1 |
| Photometric | Co-occurrence | 6 | 5 | 6 | 3 | 2 |
| | Run-length | 6 | 6 | 7 | 3 | 2 |

Table 7.16 Parameters of the selected experiments from group-D (epoch = 1000, max_fail = Inf)

| | mu_dec | mu_inc | mu_max | elapsed time (sec.) |
|---|---|---|---|---|
| G5 (Exp-16) | 0.5 | 10 | 100 | 432 |
| G6 (Exp-1) | 0.05 | 5 | 100 | 130 |

Figure 7.13 and Figure 7.14 depict overall classification performance of the Decision Engine used in G5 and G6 experiments, respectively. As seen from the figures, performance of the engine is not affected so much for more than half of the parameter setting except the non-nodule detection performance is affected slightly less than nodule detection by variation of parameters for G6; other than that, G5 and G6 show a similar response against parameter variations.

Figure 7.15 - Figure 7.19 portray an overall comparison of group-A to group-D experiments based on sensitivity, type II error (*GCFN*), specificity, and type I error (*GCFP*) performances. All experiment sub-groups produced the perfect sensitivity score at the final layer of the engine while G3_1, G6, G5 and G1, G2, and G3 yield scores between 70 and 80 in the order at the first layer of the engine (Figure 7.13). The type II error performance is almost the same (Figure 7.14). When specificity (i.e. non-nodule detection performance) and type I error performance were considered, G3_1, G3, G1 produced the perfect performance while all others reached 88%, as seen in Figure 7.15 and Figure 7.16. The accuracy performance of predictors at the final layer is shown in Figure 7.17; G3_1 and G1 produced the perfect output value, they were followed by the G5, G6 G3, and G2 in the order. We can conclude from these results that 1) allowing a longer training time can increase performance (as in G3_1), 2) a single layer

hidden network with enough nodes can produce a perfect score (as in G2) with the early stop training, and 3) adding some extra hidden layers may not yield a perfect result when the training goal is ‘continue until one of the performance value is met’.

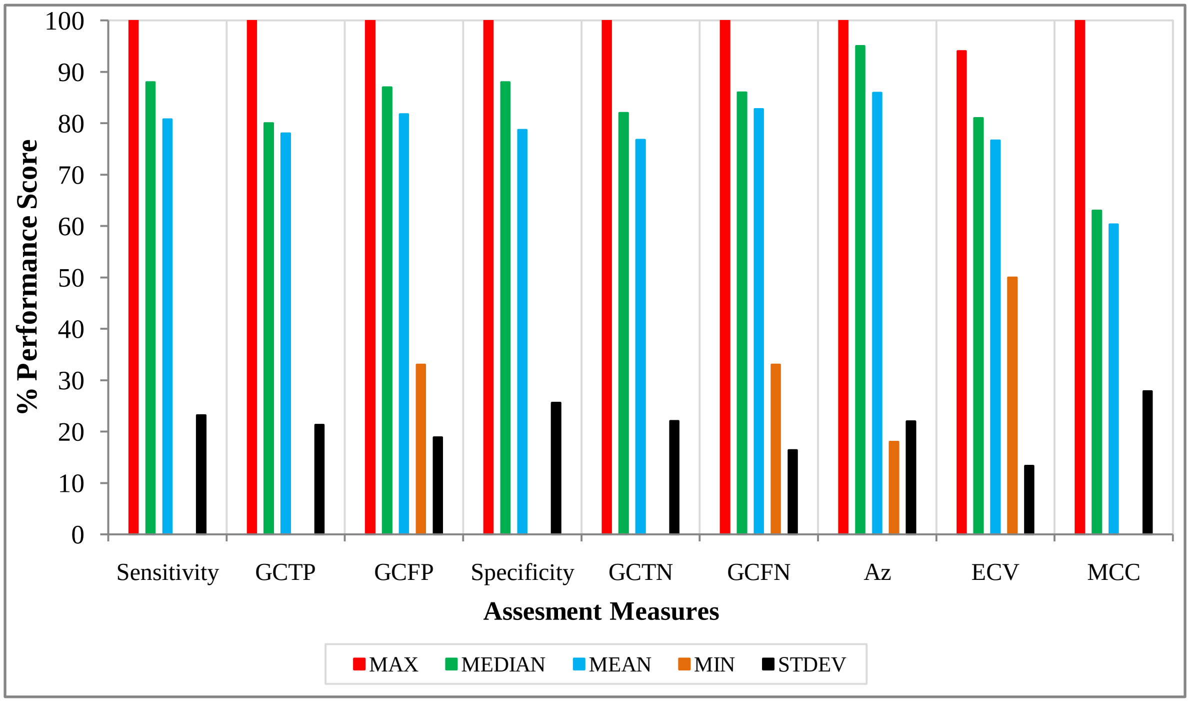


Figure 7.13 Overall classification performance of the Decision Engine against parameter variations (summary of all G5 experiments)

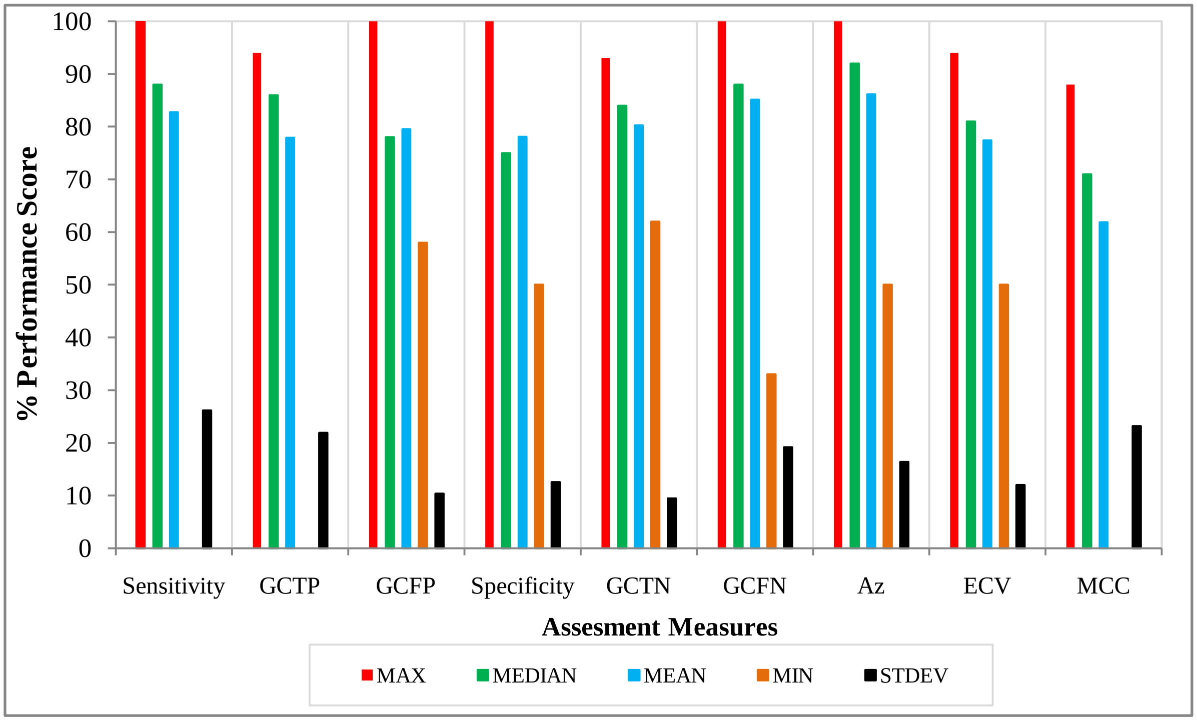


Figure 7.14 Overall classification performance of the Decision Engine against parameter variations (summary of all G6 experiments)

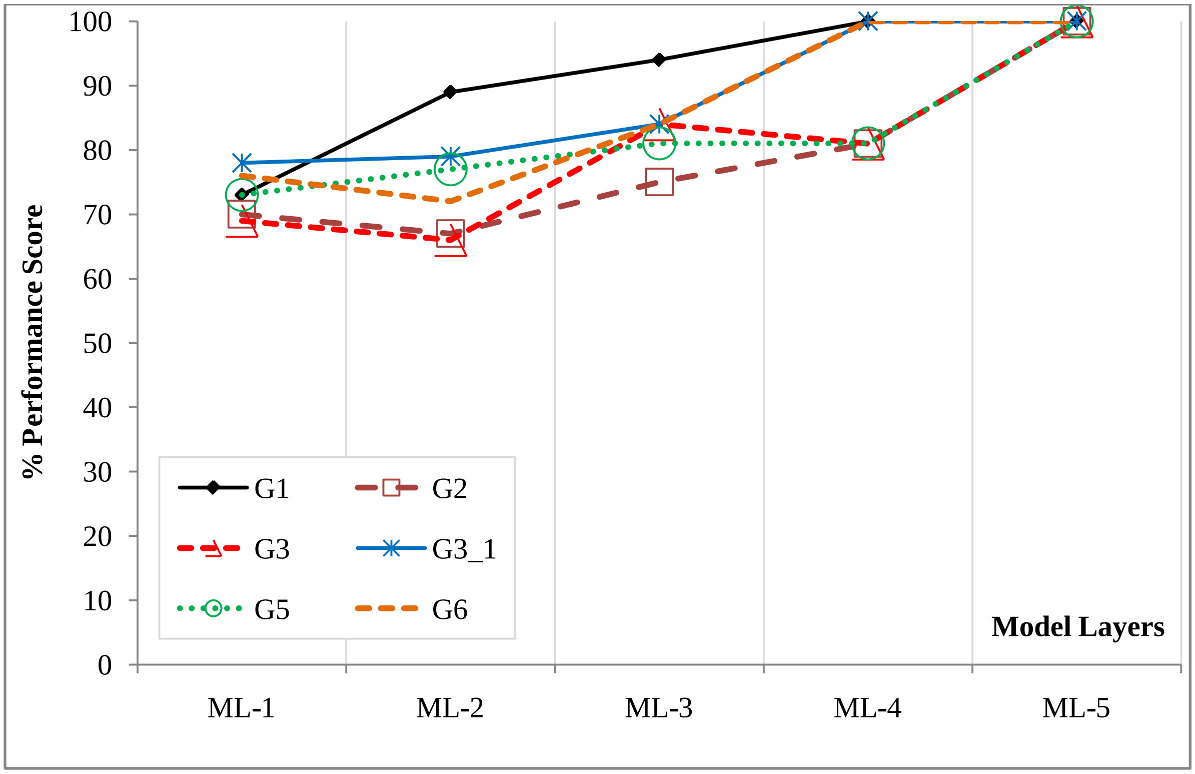


Figure 7.15 An overall comparison of the experiment sub-groups based on Sensitivity

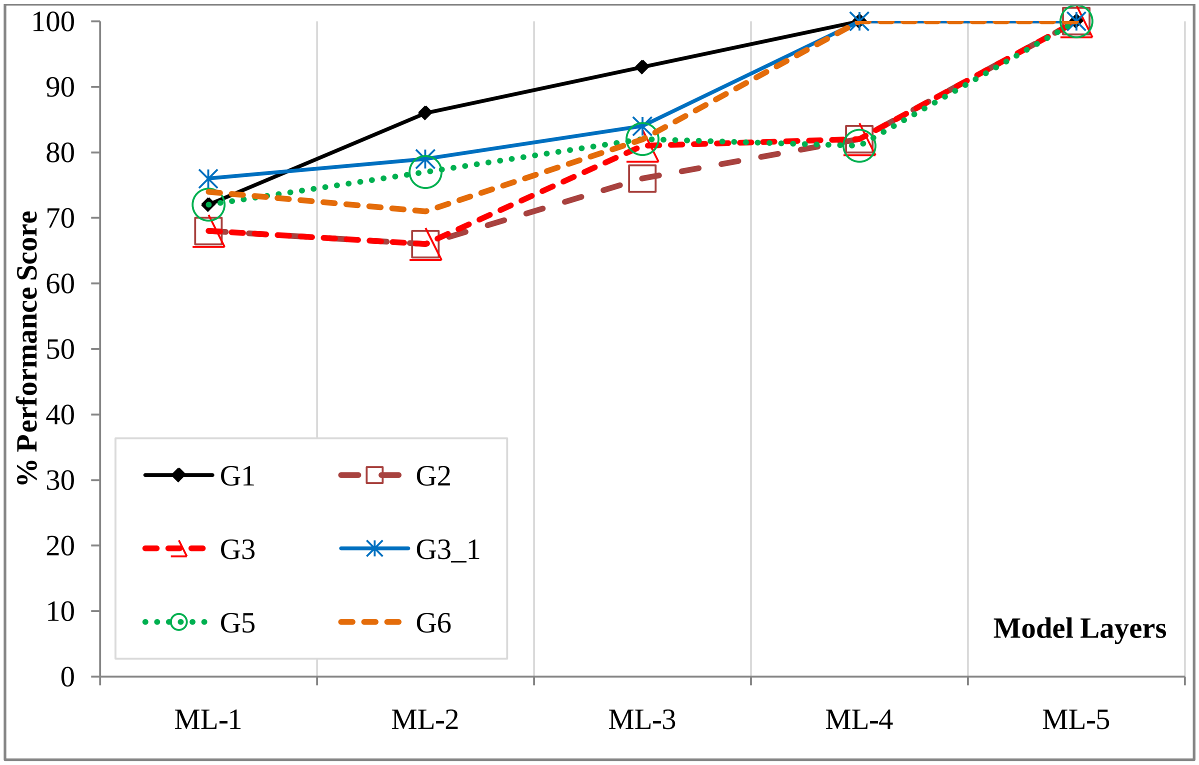


Figure 7.16 An overall comparison of the experiment sub-groups based on *GCFN*

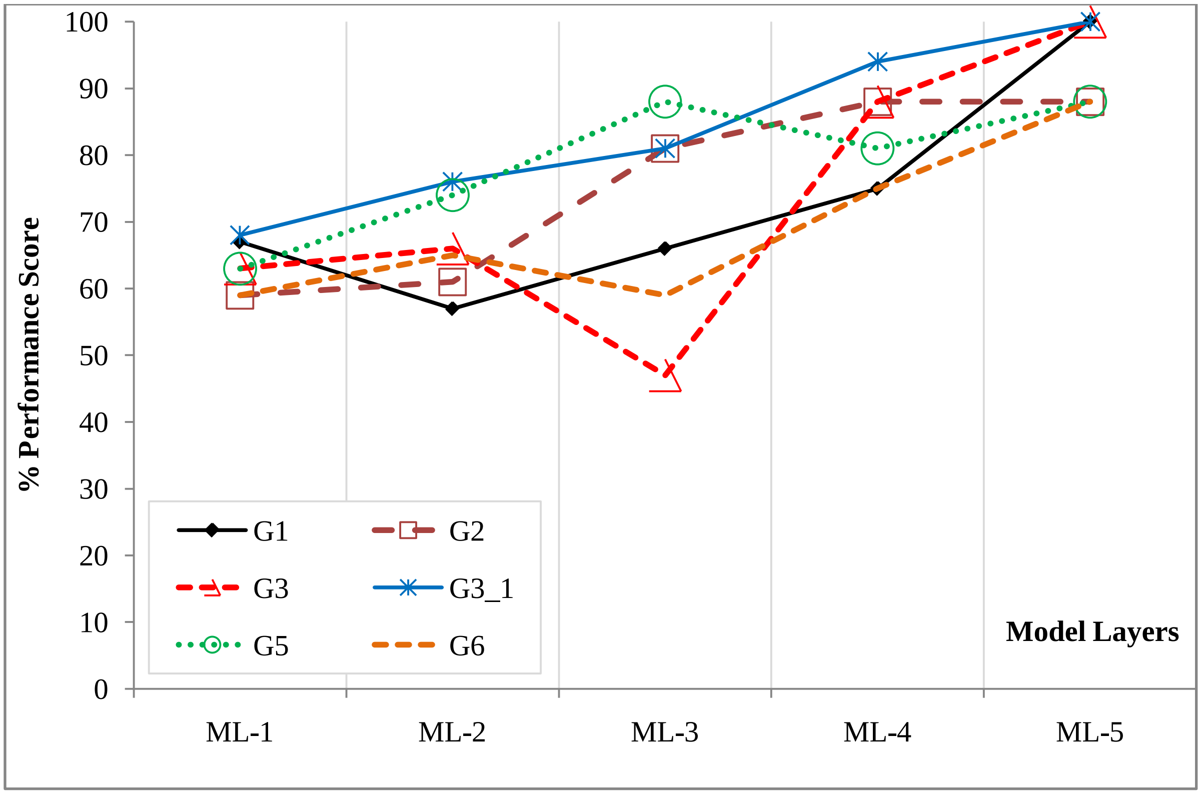


Figure 7.17 An overall comparison of the experiment sub-groups based on  Specificity

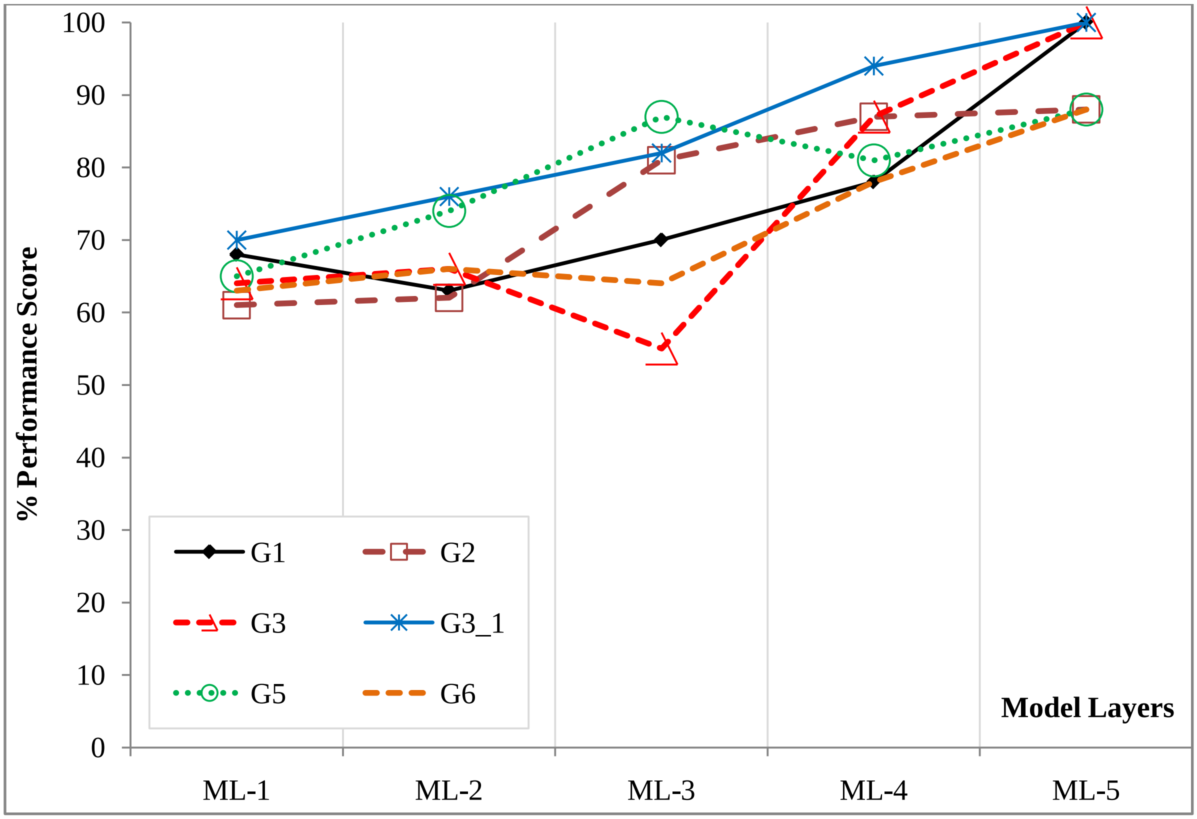


Figure 7.18 An overall comparison of the experiment sub-groups based on  *GCFP*

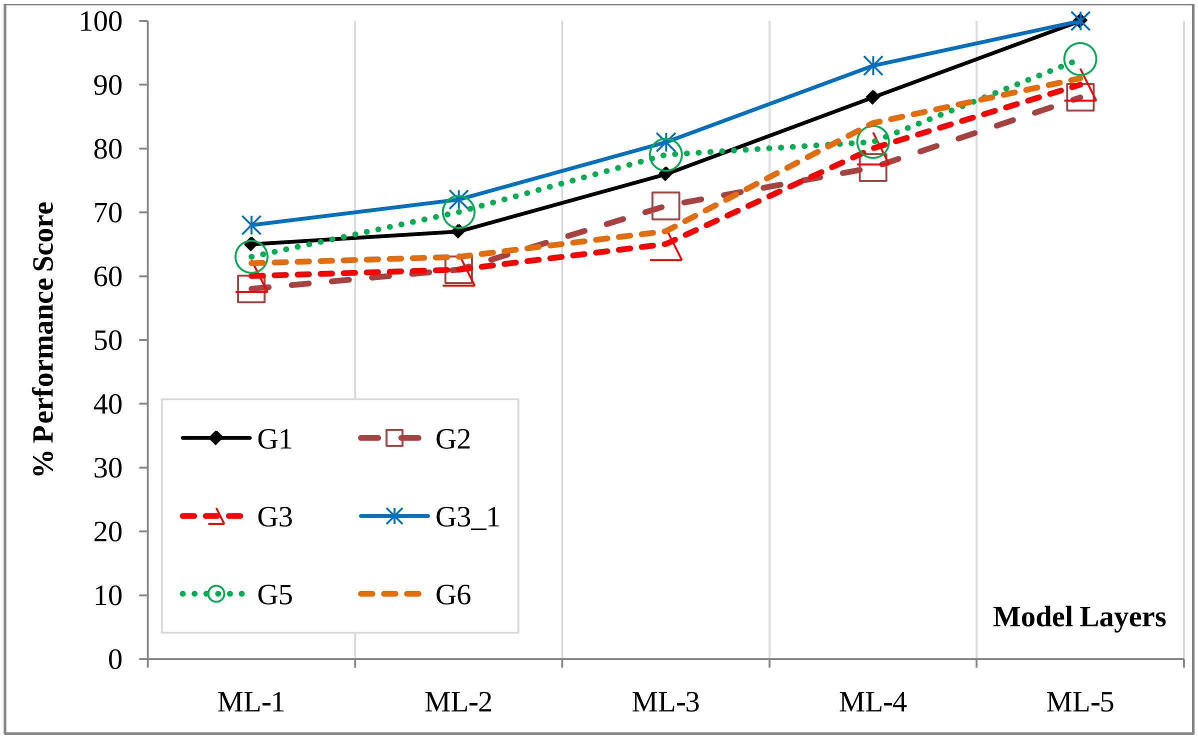


Figure 7.19 An overall comparison of the experiment sub-groups based on *ECV*

## 7.5) Experiment Group E

The experiment group-E includes several sub groups of experiments utilizing a heuristic method that combine adaptive learning rate with momentum. Each sub-group has 72 experiments with the parameters tested lr = {0.01, 0.05, 1}, lr_dec = {0.35, .7}, lr_inc = {1.05, 3.15}, max_perf_inc = {1.05, 3.15}, mc = {0.25, 0.5, .9}, max_fail = Inf, and epochs = 1000. Experiment sub-groups differ from each other based on their network structure; the output layer of each ANN has 1 node. These sub-groups are 1) the G1 has 1 hidden layer and the number of nodes in the hidden layer of each predictor is given in Table 7.1, 2) the G2 has 1 hidden layer with 1 node in each, 3) the G3_1 has 1 hidden layer with 1 node in each except the first layer whose number of nodes is the same with G1's first model layer ({2, 9, 11, 12}), 4) G5 has two hidden layers for each ANN; the number of nodes are given in the Table 7.15, and 5) G6 has three hidden layers at the first model layer of the Decision Engine. The parameters of the experiments which produced the best performance are given in Figure 7.17.

Table 7.17 Parameters of the selected experiments from group-E (max_fail = Inf, epoch = 1000)

| | lr | lr_dec | lr_inc | max_perf_inc | mc | elapsed time (sec.) |
|---|---|---|---|---|---|---|
| G1 (Exp-38) | 0.05 | 0.7 | 1.05 | 1.05 | 0.5 | 165 |
| G2 (Exp-25) | 0.05 | 0.35 | 1.05 | 1.05 | 0.25 | 248 |
| G3_1 (Exp-39) | 0.05 | 0.7 | 1.05 | 1.05 | 0.9 | 244 |
| G5 (Exp-16) | 0.01 | 0.7 | 1.05 | 3.15 | 0.25 | 91 |
| G6 (Exp-1) | 0.01 | 0.7 | 1.05 | 3.15 | 0.9 | 124 |

The response of the Decision Engine against parameter variations are depicted in Figure 7.20 - Figure 7.24. As seen, in all experiment groups, half of the parameter sets yield a performance above 75% (sensitivity) and 76% (type II error performance).

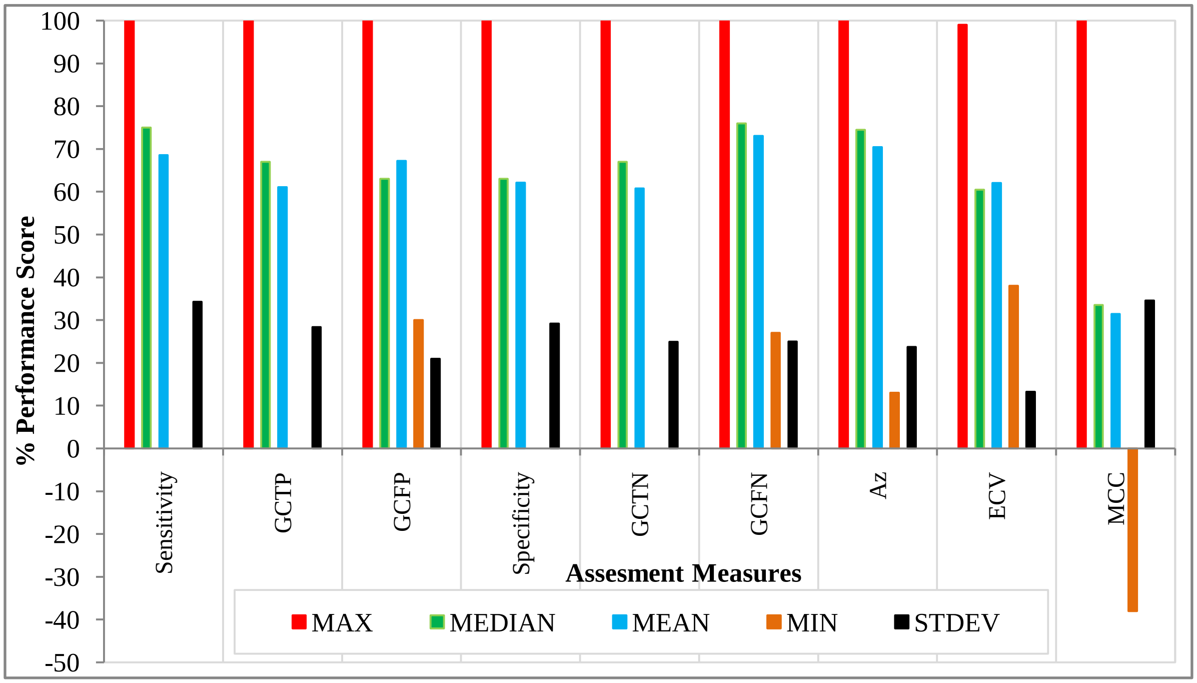


Figure 7.20 Overall classification performance of the Decision Engine against parameter variations (summary of all G1 experiments)

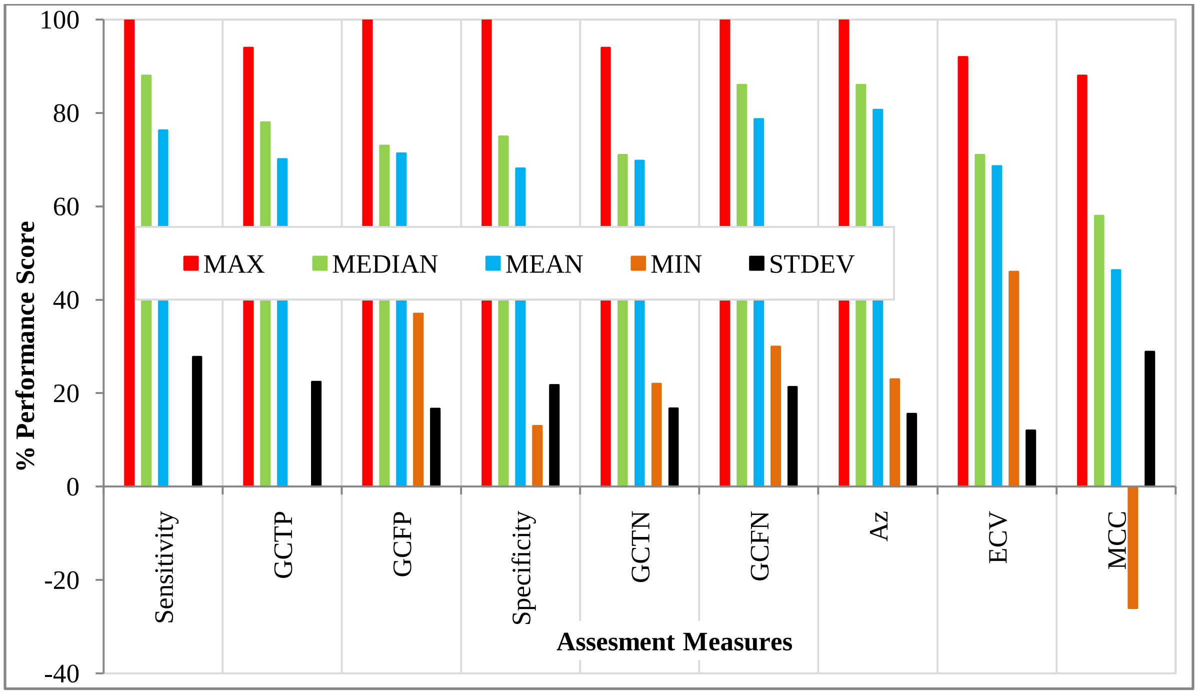


Figure 7.21 Overall classification performance of the Decision Engine against parameter variations (summary of all G2 experiments)

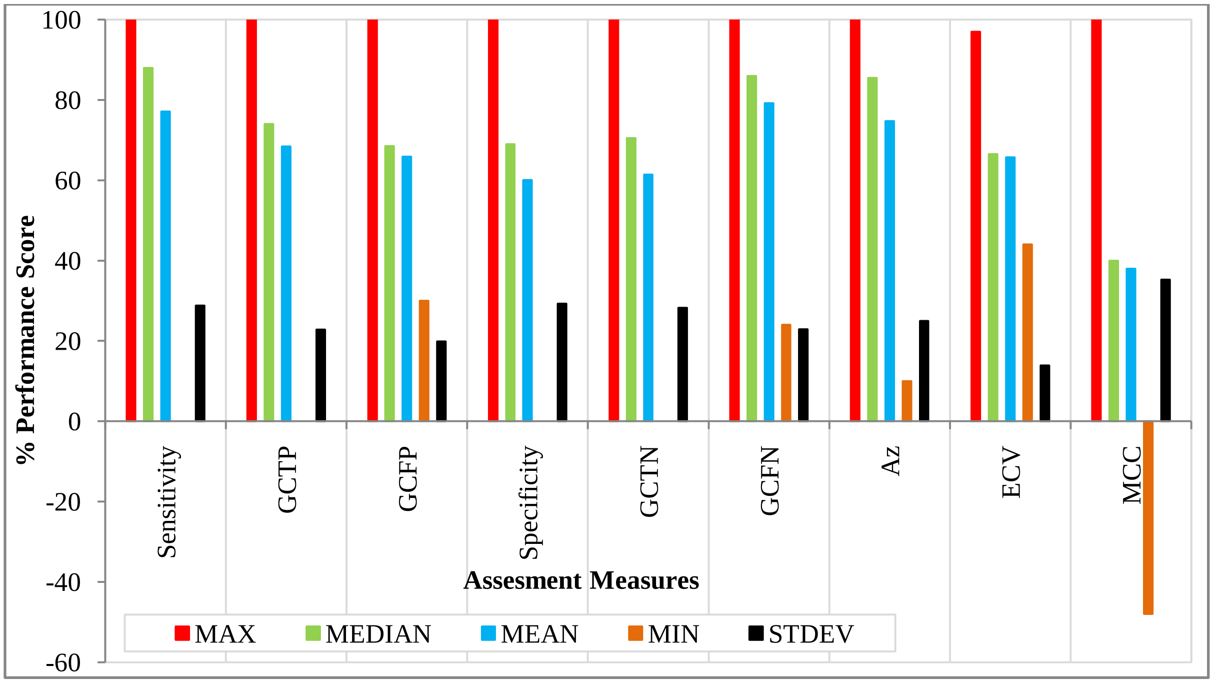


Figure 7.22 Overall classification performance of the Decision Engine against parameter variations (summary of all G3_1 experiments)

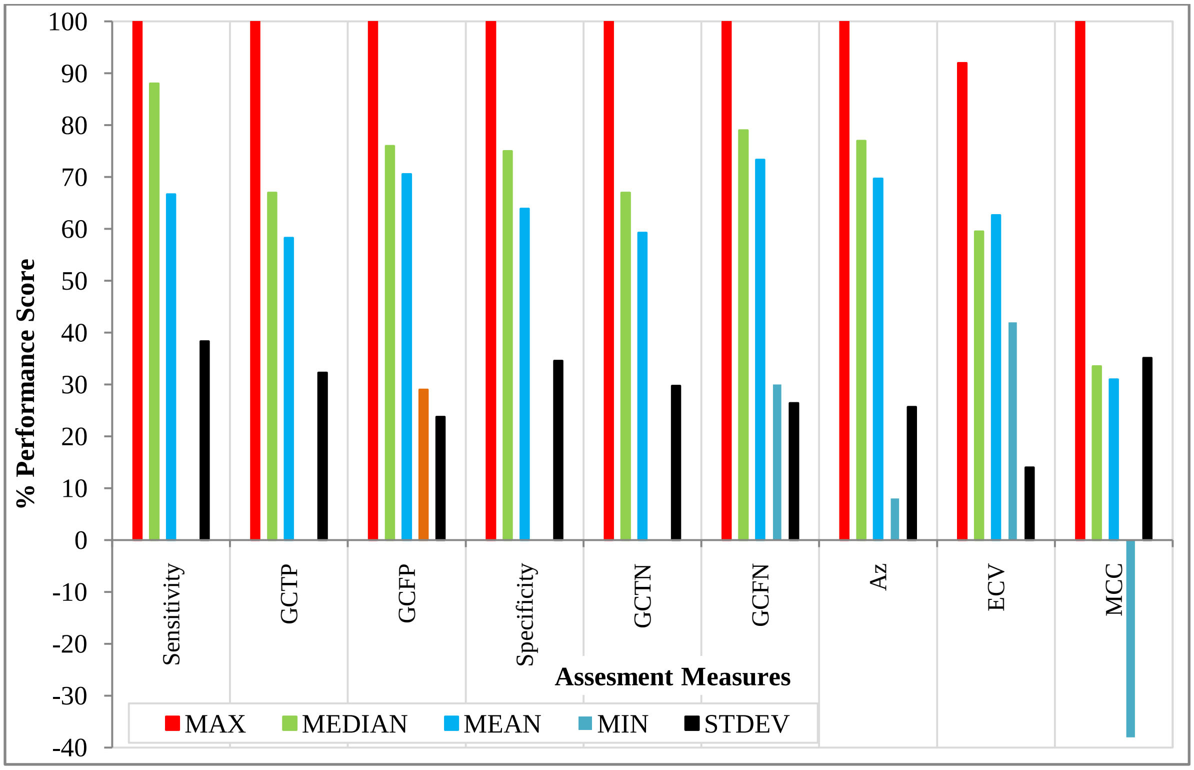


Figure 7.23 Overall classification performance of the Decision Engine against parameter variations (summary of all G5 experiments)

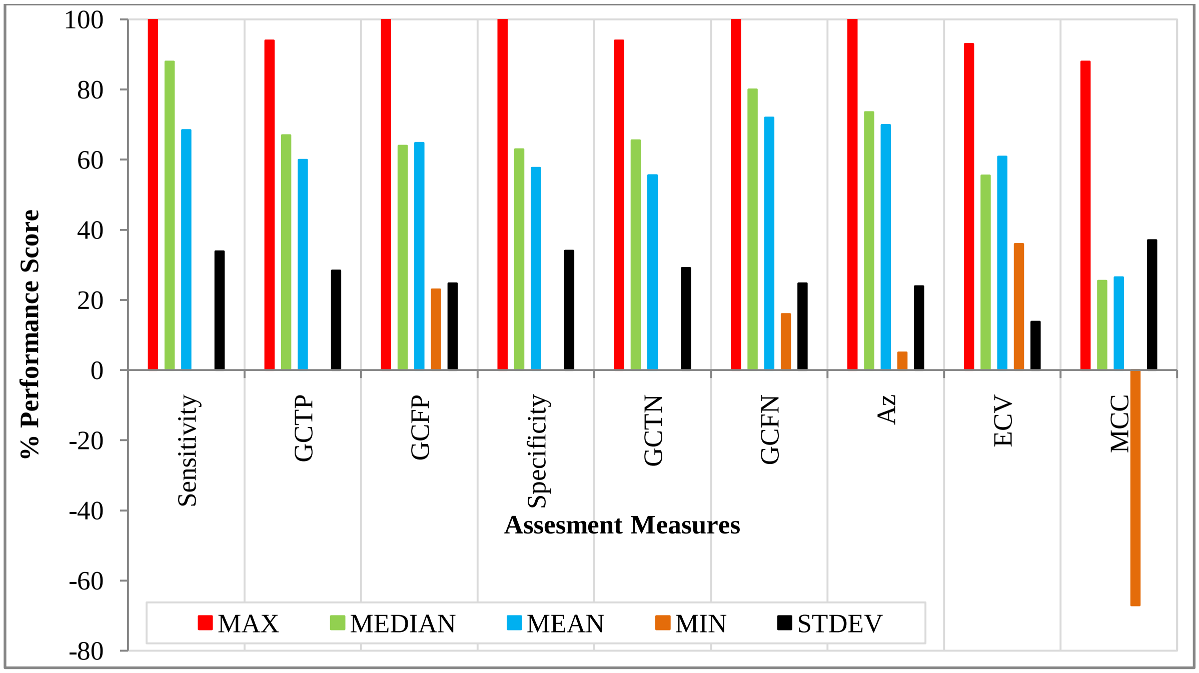


Figure 7.24 Overall classification performance of the Decision Engine against parameter variations (summary of all G6 experiments)

As seen in the Figure 7.25, the comparison of experiment sub-groups based on sensitivity shows that all of them except G6 started with a prediction performance around 70% at the model layer 1, on the other hand all sub-groups reached the perfect classification score. Among these groups, the G3_1 inclined quickly to the perfect score; it was followed by G1, G2, G5, and G6 in the order. The *GCFP* response was similar to the sensitivity as depicted in the Figure 7.26. According to the specificity and *GCFP* scores, G1, G3_1, and G5 reached the perfect score while G2 and G6 reached 88% at the final layer, as seen in Figure 7.27 and Figure 7.28. The average accuracy of the prediction outputs is depicted in the Figure 7.29; as seen, G1 accomplished the perfect score and it was followed by G3_1 closely (97%), and then G2, G5, and G6 come in the order.

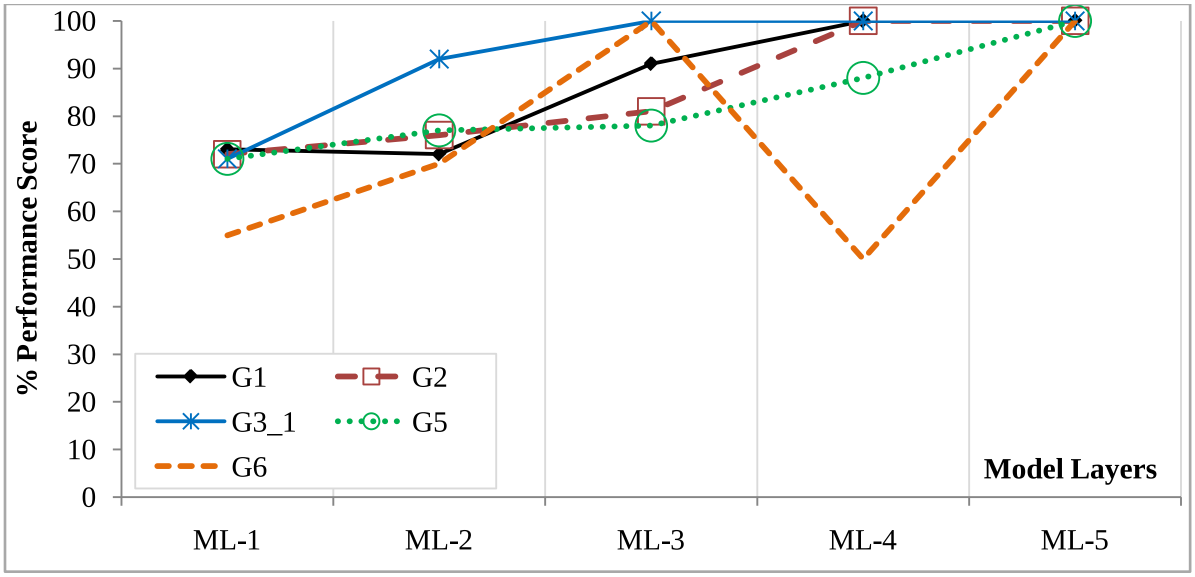


Figure 7.25 An overall comparison of the experiment sub-groups based on  Sensitivity

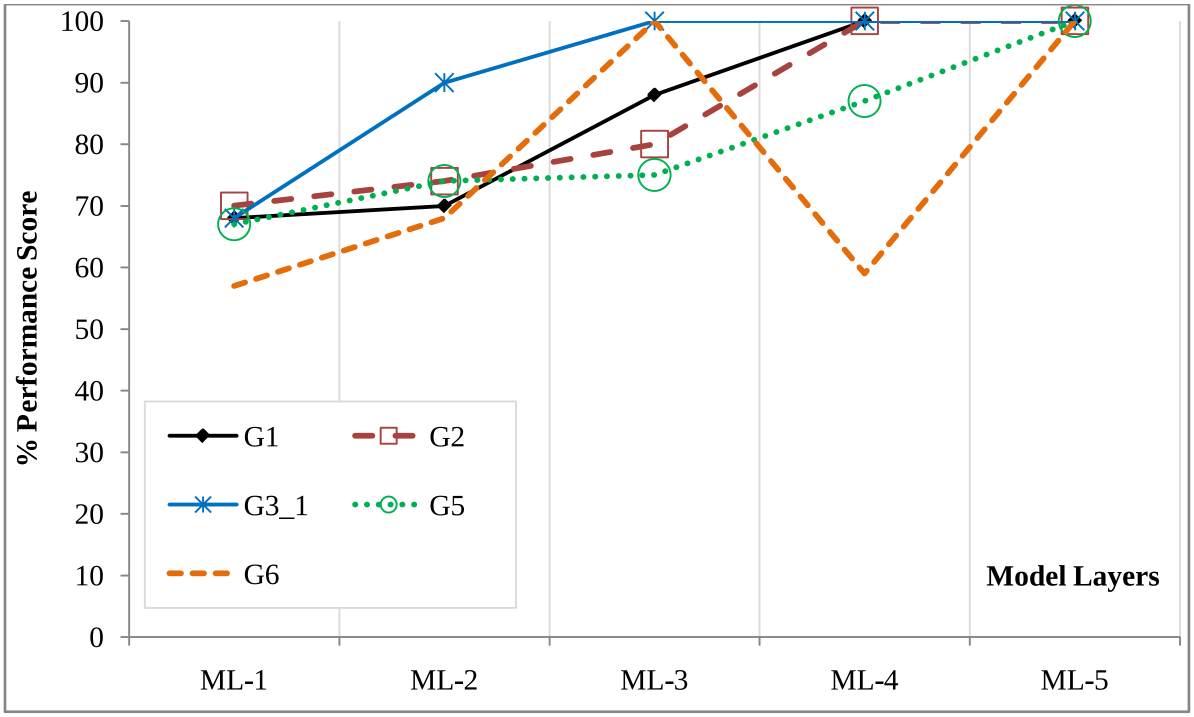


Figure 7.26 An overall comparison of the experiment sub-groups based on *GCFN*

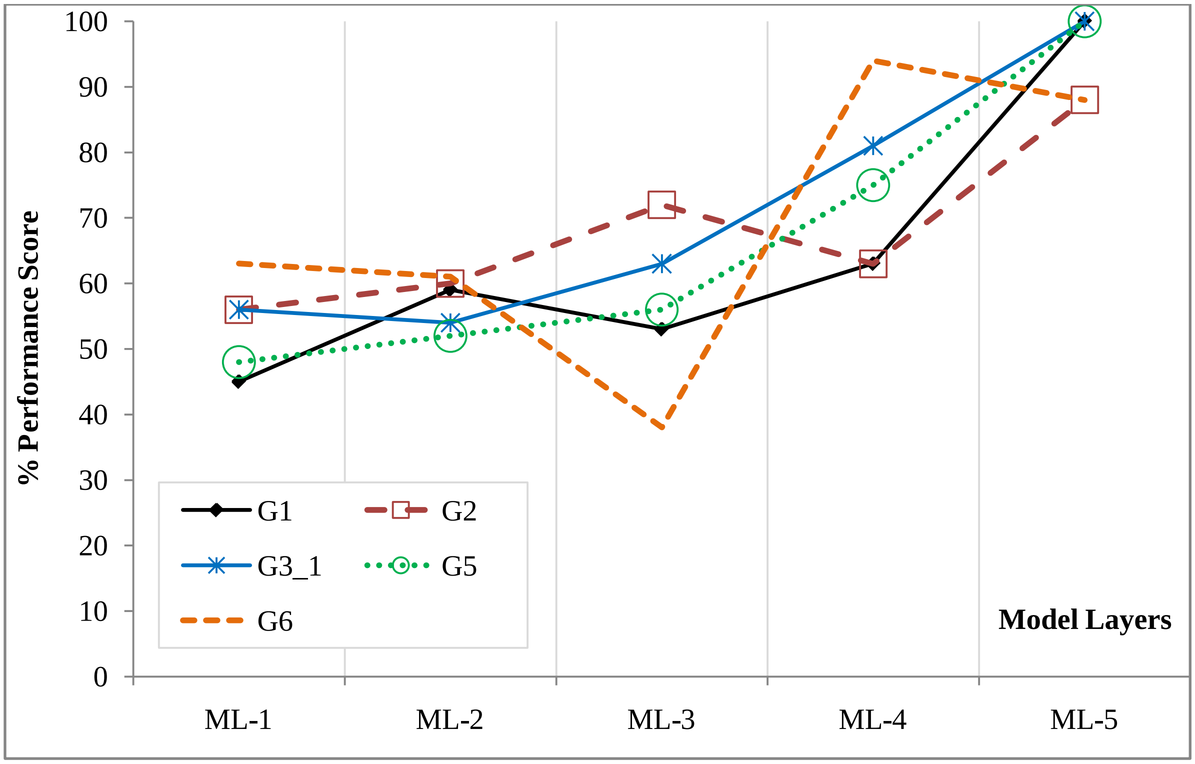


Figure 7.27 An overall comparison of the experiment sub-groups based on Specificity

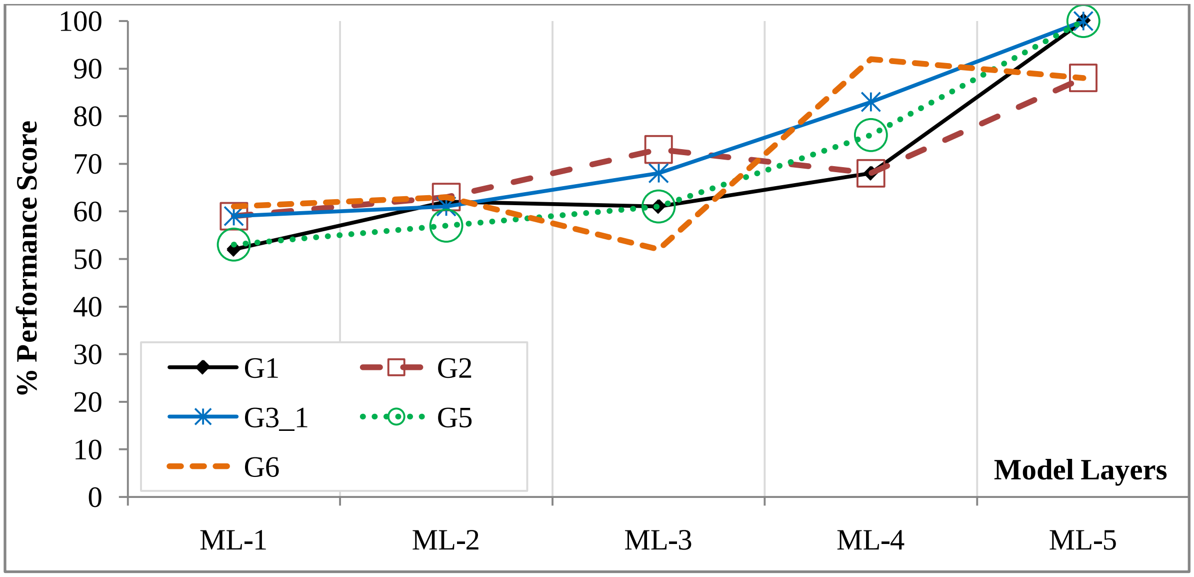


Figure 7.28 An overall comparison of the experiment sub-groups based on *GCFP*

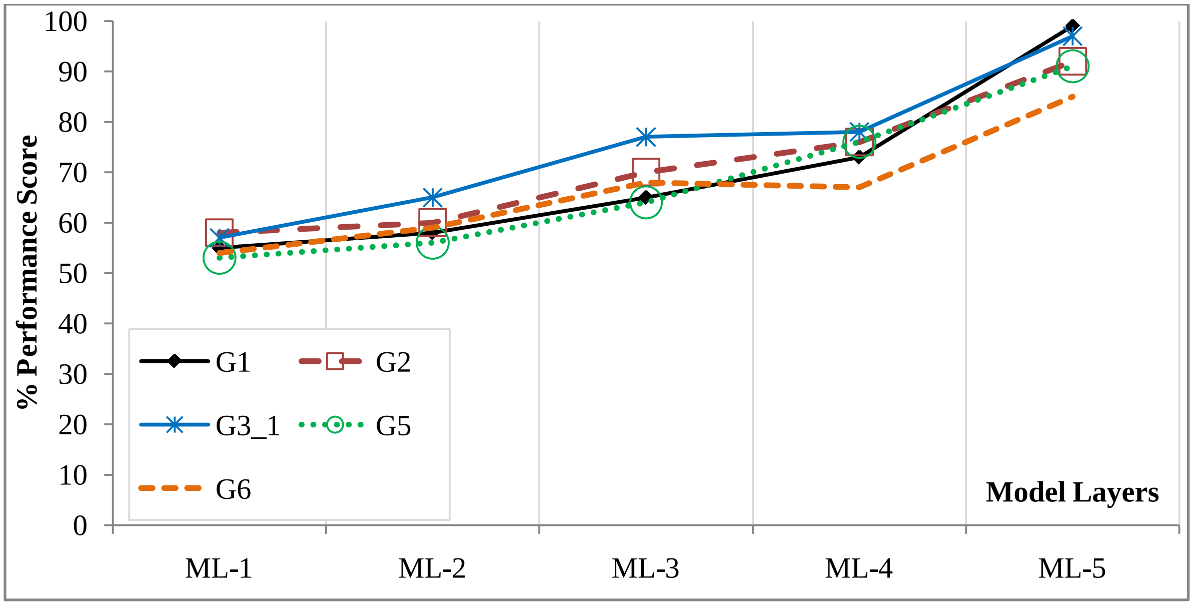


Figure 7.29 An overall comparison of the experiment sub-groups based on *ECV*

## 7.6) Experiment Group F

The experiment group F is conducted to explore how good is the proposed method over different test data. We used k-fold cross validation with *k* = 5 and run the decision engine for using the network structure and parameters that are used in the experiment G1. Among 72 experiments, the experiment 26 parameters, which are epochs = 50, mu_dec = 0.1, mu_inc = 5, and mu_max = 1000, produce a good result. The summary and classification performance of 5-runs at the model layer 5 for the experiment 26 are seen in Figure 7.30 and Figure 7.31, respectively. As seen from the figures, the proposed model yields high performance for different test data sets; this leads us to conclude that the modular decision engine is a very promising classifier.

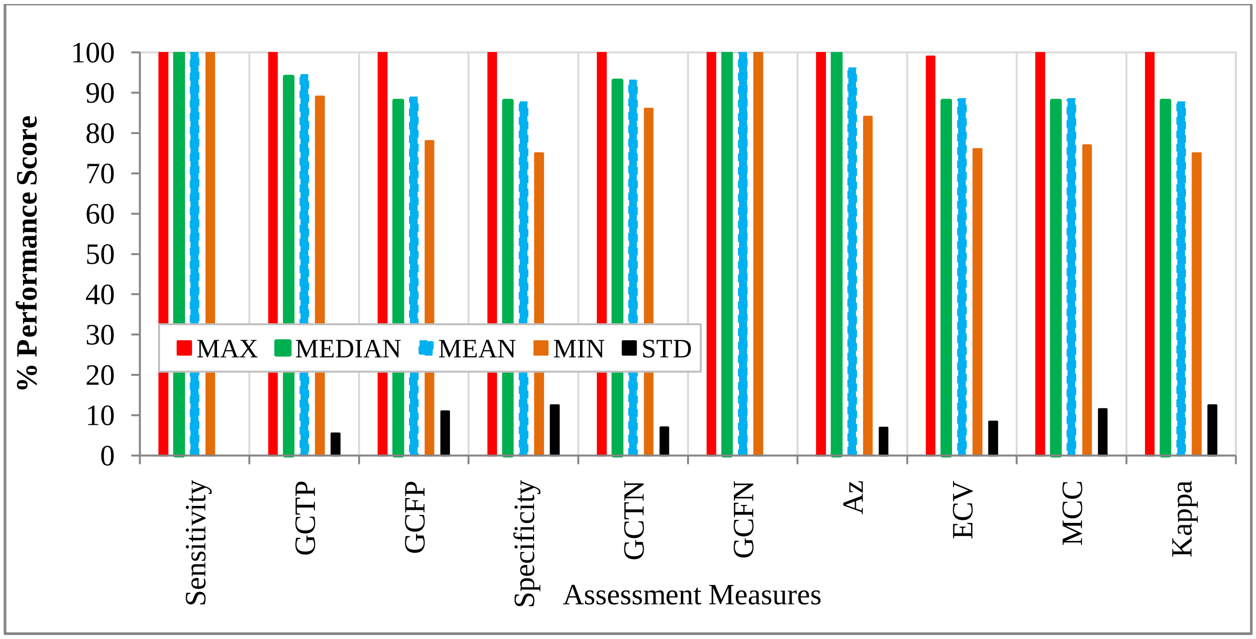


Figure 7.30 Classification Performance (summary of all runs, Exp-26)

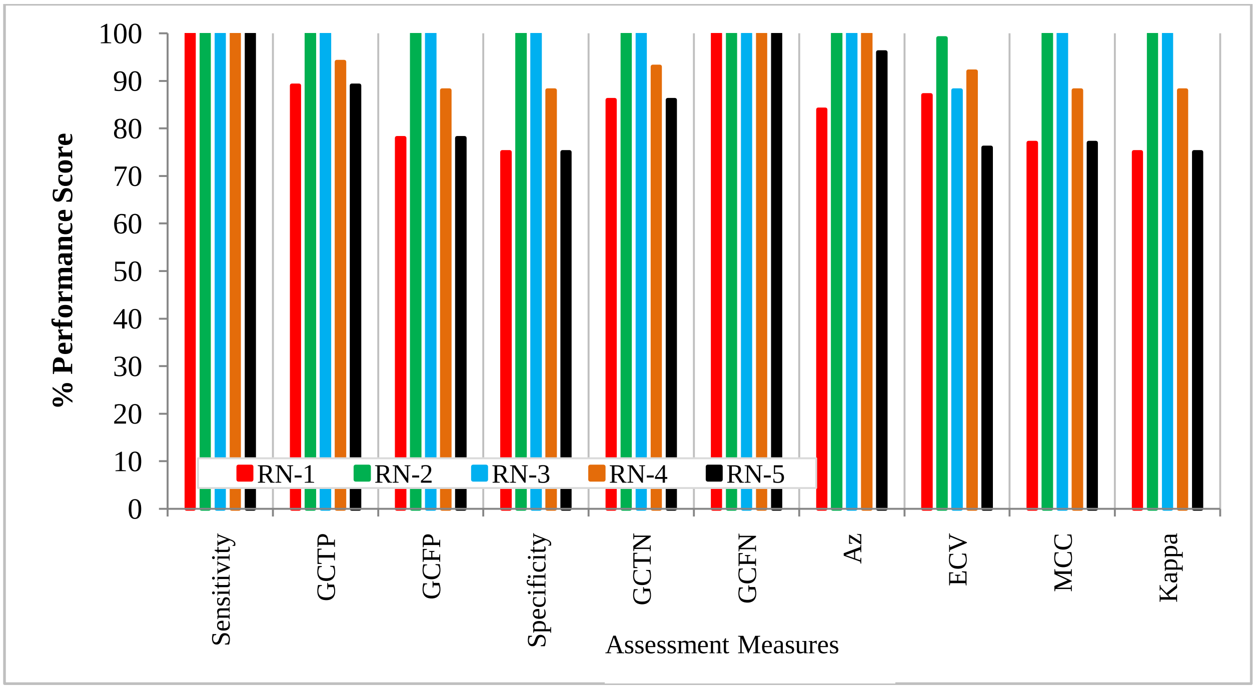


Figure 7.31 Classification Performance (comparison of runs, at model layer 5, Exp-26)

## 7.7) Comparison with Relevant Studies

An extensive literature review of the nodule detection methods are introduced in the Section (2.4). We summarize them here including their assessment scores to compare our results with them. All of them used their own data set except [Way2006], which used LIDC (Lung Imaging Database Consortium) database. The assessment measures used by these studies were area Az under the ROC curve and/or sensitivity with FP (false positive) rate. The Table 7.18 shows a list of some nodule detection methods with their performance.

Table 7.18 Comparison of Lung Nodule Detection Methods (Az: Area under the ROC curve, FP: False positive, FPC: FP candidate, T2: Type-II error, kfv: k-fold cross validation, loo: Leave-one-out, VAL.: Validation, nm: Not mentioned, REF: Reference)

| METHOD | YEAR | # NODULE (Non-Nodule) | ASSESMENT MEASURE | ASSESMENT SCORE | VAL. | REF. |
|---|---|---|---|---|---|---|
| Proposed modular method | 2009 | 8 (8) | Average Sensitivity, specificity, Type I error, Type II error, Az, etc. | 100%,<br>88%,<br>88.8%,<br>100%,<br>96% | kfv (*k*=5) | This dissertation |
| k-means clustering followed by 3D active contour | 2006 | 23 | Az | 83% ±4 | loo | Way2006 |
| Threshold followed by LDA | 2005 | 31 | Az | 85.6% w/ 9.5 FP/case (0.04 FP/image) | loo | Zhang05 |
| Ellipse fit + LDA | 2005 | 116 | Sensitivity | 80% w/ 0.37 FP/section | loo | Ge2005 |
| Gray-level threshold and region growing, statistical shape model | 2004 | - | Sensitivity | 90% w/ 5.6 FP/date set | loo | Paik2004 |
| Massive Training Artificial Neural Network | 2004 | 109; 38 | Sensitivity | 83%, 5.8 FP/scan; 84%, 5.9 FP/scan | nm | Arim04 |
| Threshold followed by LDA | 2003 | 470 (212,236) | Az; FPC | 79%; 15,266 | loo | Arma03 |
| Massive Training Artificial Neural Network | 2003 | 71 (1726) | Sensitivity | 80.3, 4.8 FP/patient (0.18 FP/section) | nm | Suzuki03 |
| Weighted k-means clustering and manual segmentation | 2002 | 63 (>8,000) | Sensitivity | 84% w/ 1.74 FP/section | nm | Gurc02 |
| Volume projection | 2002 | 34; 32 | Sensitivity | 67.6%; 71.9 % w/ FP/slice (for both) | nm | Wei2002 |
| Threshold followed by LDA | 2001 | 20 to 40 | Sensitivity; Az | 70%; 90% | nm | Arma01 |
| GA template matching algorithm | 2001 | 98 (88,318) | Sensitivity | 72% with 1.1 FP/section | nm | Lee001 |
| rule based followed by LDA | 2001 | 31 (>5344) | Sensitivity | 84% w/ 1.75 FP/slice | nm | Gurc01 |
| Threshold + LDA | 2000 | 300 (27,917) | Az; T2; FPC | 93%; 4.4%; 65.8% | nm | Arma03 |
| 3D deformable surface model approach + LDA | 2000 | 128 | Az | 97% | loo | Kawa00 |
| A 3D region growing is employed + rule based | 1999 | 23 | Sensitivity | 100% for size ≥ 5 mm (n=10) and 95.6% for size ≥ 3 mm (n=23) with 15 FP/study (< 1 per image) | nm | Fieb99 |
| Threshold + circularity | 1999 | 18 | Sensitivity; Specificity | 89.9%; 11.1% | nm | Lou99 |
| Threshold followed by LDA | 1999 | 187 | Sensitivity; Az | 72% w/ 4.6 FP/section; 93% | loo | Arma99 |
| Segmentation using 3D geometric approach + LDA | 1999 | 128 | Az | 92% | loo | Kawa99 |
| VNQ-filter | 1998 | 21 | Sensitivity | 100% w/ 93.5% FP | nm | Okumur98 |
| A tree based on four threshold | 1994 | - | Sensitivity | 94% w/ 1.25 FP/case | nm | Gige94 |

An exact comparison with these studies would be difficult; because the data sets used are not the same. Another difficulty in comparison is that most of the studies have reported their assessment results based on sensitivity with FP rate/section (case) and Az while the ratio of positive class samples to negative class samples is so low. This kind of assessment would be misleading because a classifier that tends to classify samples as ‘nodule’ will have a higher FP rate. On the other hand, FP rates are given as either per section (slice, image) or per case (scan) and they sometimes report the total number of non-nodules (FP candidates); this makes more difficult to compare specificities. In addition, almost all studies did not mention their type II error performance, which is crucial to evaluate a classifier for nodule detection due to cost of a life. Also, current research reports did not mentioned how their approaches are sensitive to parameter variations.

We used 80 imjects (image objects) of 40 nodules and 40 non-nodules. The nodule samples were extracted from 20 patients’ image series provided by National Cancer Institute from National Cancer Imaging Archive and non-nodule samples were extracted from a healthy person’s image series, which were provided by [LakePet]. We tested our model employing six experiment groups; each group has several sub-groups. In each sub-group, 72 experiments were conducted except in two sub-groups, which have 18 experiment combinations.

In this dissertation, we used several assessment measures to evaluate our system from different aspects, especially, including type I and type II error assessment. In addition, we have experimented with many combinations of parameters for different network structures to test effect of parameter variations on the performance. The results show that most of the parameter sets (at least more than half of the experiment cases) are able to produce very good performance while it is possible to get so weak performance for some parameter sets. Test results show that, the proposed modular model can produce a perfect classification as regard to nodule detection, non-nodule detection, type I and type II error performance measures.

As regard to multi-perspective analysis, current approaches perform 3D detection through transverse plane. Most of the studies did not mention their training times except [Fieb99] where the computation time is reported as 5 min using an O2 workstation of 250 MHz for detection of 23 nodules. In [Lee001], GA template matching is used for detection. It takes 23 min. The same authors used a template matching algorithm alone; they reported that it takes 164 min. for detection by ANN. Its training takes 29.9 hours with Pentium 1.7 GHz PC-based workstation. In our detection scheme, the min and max training time of the best parameter set observed for Levenberg-Marquet approach are 8 sec (G1) and 432 sec (G5), respectively; they are 91 sec (G5) and 248 sec (G2) for variable learning rate with momentum approach, respectively. We used Intel workstation dual core PC of 3.20 GHz and 3.19 GHz with 3.9 GB of ram.

There are so many feature vectors used in the current literature as reviewed in the Section 2.5. We used four feature vector types, namely spectral shape feature, which we derived from our canonical representation, circularity-topology-compactness features, co-occurrence based features, and run-length based features. Our spectral feature vector performance is around 80%; this score may go higher if one uses scale invariance as we discussed in Section 4.3.

# CHAPTER 8. CONCLUSION AND FUTURE DIRECTIONS

Lung cancer is one of the most lethal cancer types. Researches in computer aided detection (CAD) and diagnosis (CADD) for lung cancer aim at providing effective tools to assist physicians in cancer diagnosis and treatment to save lives. The current imaging technology in medicine produces huge amount of images to be analyzed by radiologists. It has been reported that radiologist may overlook some nodules due to heavy load of number of images and fatigue. A CADD system can decrease the amount of load from a radiologist as well as can provide a 'second opinion' to increase their interpretation performance.

## 8.1) Contributions

In this dissertation, our main goal is to develop a comprehensive CAD framework encompassesing a state of the art decision engine model that mimics the way human experts approach the nodule detection task, combines spatial higher dimensional analysis with multi-type feature analysis, accommodates multi-learning scheme, allows flexibility to be modified without affecting entire system, suits to parallel processing implementation, and be able to accomplish all these capabilities with demanding less computational resources and efficient processing.

As the central part of the proposed framework, we develop a novel hierarchical modular decision engine implemented by ANN. The second main contribution of this dissertation is developing a novel shape representation technique that can be employed in pattern recognition and retrieval applications. In addition, we proposed several shape feature derived from the proposed shape representation.

The proposed framework is composed of several integrated units including data retrieval, pre-processing, feature extraction, and modular decision engine for pattern classification. The data retrieval unit is responsible mainly from creation of VOIs to be used in training and testing. The feature extraction unit computes feature vectors from VOIs and organizes them for modular multi-level decision process. The modular decision engine that encompasses important properties mentioned above performs classification based on spatial and feature level analysis and synthesis.

We proposed three modular hierarchical decision engine models; these models are Inter-Slice Inter-Plane (ISIP) model, Intra-Slice Intra-Plane (ISAP) model, and Intra-Slice Intra-Plane (ASAP) model. The intuition behind the proposed models is mimicking a radiologist's image reading technique; besides, taking advantage of machine intelligence that can perform beyond the human vision limited capacity.

We implemented ISIP model by ANN with Backpropagation learning technique. The model is assembled by five layers and its each layer has several modules that are composed of a predictor and a regrouping processor. We tested this model utilizing two Backpropagation learning algorithms, which are Levenberg-Marquet and variable learning rate with momentum.

The data model of the modular engine differs from that of non-modular classifier in such a way that all of the volumetric data is utilized together to obtain a decision while the modular approach partitions the raw data into several spatial and feature analysis-synthesis levels. Therefore, the proposed modular engine model mimics a human expert who analyzes a VOI from different perspectives; further, the engine, after obtaining machine coded features from the raw data, enhances the decision-making process with multi-level feature analysis which is beyond the limit of a human intelligence.

An efficient algorithm is developed to compute the new polygonal shape representation proposed, namely ‘canonical polygonal representation’ (CPR), and its normalized version ‘normalized canonical polygonal representation’ (NCPR). The NCPR can provide a unique representation of a polygon under rigid transformation and can be applied to affine transformation with some additional processes. We derived three shape features, which are ‘spectral shape feature’ and two circularity features, from this canonical representation to be used in implementation.

Another contribution of this dissertation is development of a fully-automated polygonalization algorithm that allows a user to select how smooth the boundary will be represented by a polygon.

As a part of the proposed framework, we developed several tools to automate the entire system such as data retrieval tool and feature extraction tool. In addition, we developed image data visualization and processing tool SHAHIN that is integrated with a user-interface to analyze, test, and extract VOIs.

Assessment of the method is performed using several measures for different training models and parameters. These assessment methods are sensitivity, goodness of classification (GC) based on true positive rate, GC based on false positive rate, specificity, GC based on true negative rate, GC based on false negative rate, area under the ROC curve, Matthews Correlation Coefficient, and expected closeness value.

We used 80 imjects (image objects) of 40 nodules and 40 non-nodules. The nodule samples were extracted from 20 patients’ image series provided by National Cancer Institute from National Cancer Imaging Archive and non-nodule samples were extracted from a healthy person’s image series, which were provided by Lake P.E.T. Imaging Center. We tested our model employing six experiment groups; each group has several sub-groups. In each sub-group, 72 experiments were conducted except in two sub-groups, which have 18 experiment combinations.

The experimental results show that detection capability is improved at each layer and the perfect classification is achieved at the final stage for some parameter sets. In addition, these results show that the final classification is not sensitive to at least half of the parameter sets while producing a high performance result.

The min and max training time of the best parameter set observed for Levenberg-Marquet approach are 8 sec (G1) and 432 sec (G5), respectively; they are 91 sec (G5) and 248 sec (G2) for variable learning rate with momentum approach, respectively.

As a limitation of this study, the size of the testing set is small, we did not experiment effect of the propose shape features on overall detection performance, and we did not explore reason behind getting low detection performance in some parameter sets. These issues will be explored in our future research proposals.

## 8.2) Future Directions

This dissertation shows that the modular decision engine proposed is a promising CAD model. On the other hand, during our research and development we observed that there are still so many challenges and improvements are waiting to be explored and experimented.

Applying the approach proposed to computer aided diagnosis: The current framework can be used to classify benign and malignant tumors without any structural design modification; the only simple modification is to define class definitions as ‘malignant’ or ‘benign’ for positive and negative class, respectively.

Implementation of other two models proposed: ISAP and ASAP models can be implemented to explore how well they combine multi-level multi-perspective analysis and synthesis.

Multi-class pattern recognition: The modules in the current scheme are designed to predict for binary-classification. Fortunately, its nature can cope with multi-class decision systems.

Testing the model with different learning algorithms: Another interesting research challenge would be applying the proposed approach to a multi-learning scheme in which several learning algorithms, such as ANN methods like delayed networks and LVQ, statistical models like HMM, fuzzy-logic, co-operate in this modular hierarchical architecture.

Designing hardware architecture of the modular decision engine: The proposed modular scheme can be integrated into a standalone device to be used in clinical environment.

Utilizing different performance cost functions for weight adjustment: In this dissertation we used sum of square errors as the performance cost function. Performance of other cost functions such as cross-entropy error can be explored for this kind of binary classification problems.

Enhancement of the framework with tools such as automatic lung area extraction and nodule candidate region search: The current framework assumes that the candidate volume is provided; a full automated system requires lung area to be extracted from each slice and candidate regions should be found once the image series of a case is provided.

Adding new feature-types and eliminating highly correlated features: We have explored and showed that the proposed modular approach is promising with two feature-classes and two feature-types in each class. Additional feature-types can be applied over a larger data set to increase discrimination capability.

Candidate search without segmentation: An important challenge in image pattern recognition is obtaining candidate regions by segmentation which is still a challenging problem in image processing area. The complex nature of lung imposes more challenge for the segmentation. Therefore, a method which does not require segmentation will make a big practical impact.

Exploring cluster of feature vector samples: Selection of samples affects the training performance. Training samples should be able to span whole feature vector space for a better generalization. So a feature vector clustering can be applied first for obtaining a representative training and validation sets.

# REFERENCES

[Arim04] H Arimura, S Katsuragawa, K Suzuki, et al.; Computerized Scheme For Automated Detection Of Lung Nodules In Low-Dose Computed Tomography Images For Lung Cancer Screening, *Academic Radiology*, 11(6): 617-29, 2004

[Arkin91] E M Arkin, L P Chew, D P Huttenlocher, et al.; An Efficiently Computable Metric For Comparing Polygonal Shapes, *IEEE Trans. PAMI*, 13(3), 209-216, 1991

[Arma00] S G Armato, M L Giger, H Macmahon, et al.; Analysis Of A Three-Dimensional Lung Nodule Detection Method For Thoracic CT Scans, *Proceedings of SPIE, Medical Imaging,Image Processing* – v.3979: 103-109, 2000

[Arma01] S G Armato, M. L. Giger, H. MacMahon; Automated Detection Of Lung Nodules In Ct Scans: Preliminary Results, *Medical Physics*, 28(8): 1552-1561, 2001

[Arma03] S G Armato, M.B. Altman, J. Wilkie, et al.; Automated Lung Nodule Classification Following Automated Nodule Detection on CT: a serial approach, *Medical Physics*, 30(6): 1188-97, 2003

[Arma04] S G Armato, G. McLennan, MF. McNitt-Gray; Lung Image Database Consortium: Developing A Resource For The Medical Imaging Research Community, *Radiology*, 232: 739-748, 2004

[Arma99] S G Armato, M L Giger, J T Blackburn; Three-Dimensional Approach To Lung Nodule Detection In Helical CT, *Procedings SPIE, Medical Imaging, Image Processing*, v.3661: 553-559, 1999

[Awai04] K Awai, K Murao, A. Ozawa, et al.; Pulmonary Nodules At Chest CT: Effect Of Computer-Aided Diagnosis On Radiologists' Detection Performance, *Radiology*, 230(2): 347–352, 2004

[Baldi] P Baldi, S Brunak, Y Chauvin, C A F Andersen, H Nielsen; Assessing the accuracy of prediction algorithms for classification: an overview, *Bioinformatics*, 16(5): 412-424, 2000

[Bernier03] T Bernier and J A Landry; A New Method For Representing And Matching Shapes Of Natural Objects, *Pattern Recognition*, 36(8), 1711-1723, 2003

[Bish06] C M Bishop; *Pattern Recognition And Machine Learning*, Springer, p. 272-7, 2006

[Buny06] T Bunyaviroch, R E Coleman; PET Evaluation of Lung Cancer, *Journal of Nuclear Medicine*, 47(3): 451-469, 2006

[Bush01] J T Bushberg, J A Seibert, E M Leidholdt, et al.; *The Essential Physics Medical Imaging*, 2nd ed, by Lippincott Williams & Wilkins, 2001

[Cancer] *http://www.cancer.gov*

[Cangov] *http://www.cancer.gov/cancertopics/wyntk/lung/page4*

[Carr98] M J Carreira, D Cabello, M G Penedo, et al.; Computer-aided diagnoses: Automatic detection of lung nodules, Medical Physics, 25(10): 1998-2006, 1998

[Chang91] C C Chang, S M Hwang, and D J Buehrer; A Shape Recognition Scheme Based On Relative Distances Of Feature Points From The Centroid, *Pattern Recognition*, 24 (11): 1053-1063, 1991

[Chen95] J M Chen, J A Ventura; Optimization models for shape matching of nonconvex polygons, *Pattern Recognition*, 28(6), 863-877, 1995

[Chuang96] J H Chuang; A potential-based approach for shape matching and recognition, *Pattern Recognition*, 29(3), 463-470, 1996

[Demi04] C Demir, B Yener; Automated cancer diagnosis based on Histapathological Images: A systematic survey, *Technical Report, Rensselaer Polytechnic Institute, Department Of Computer Science*, TR-05-09, 2004

[Doi005] K Doi; Current status and future potential of computer aided diagnosis in medical imaging, *British Journal of Radiology*, 78: S3-s19, 2005

[Dood04] L E Dodd, R F Wagner, S G Armato, et al.; Assessment methodologies and statistical issues for computer-aided diagnosis of lung nodules in computed tomography: contemporary research topics relevant to the lung image database consortium, *Academic Radiology*, 11: 462– 475, 2004

[Duda01] R O Duda, P E Hart and D G Stork; *Pattern Classification*, Wiley Interscience, 2nd ed., p. 10, Figure 1.7, 2001

[Eras00] J J Erasmus, J E Connolly, H P McAdams, et al.; Solitary pulmonary nodules: part I. Morphologic evaluation for differentiation of benign and malignant lesions, *Radiographics*, 20(1): 43–58, 2000

[Factus] http://www.nationallungcancerpartnership.org/page.cfm?l=factsUS

[Fieb99] M Fiebich; C Wietholt, B C Renger, et al.; Automatic detection of pulmonary nodules in low-dose screening thoracic CT examinations, *Proc SPIE*, v.3661: 1434-1439, 1999

[Foldoc] *http://foldoc.org/foldoc/foldoc.cgi?pattern+recognition*

[Gallo75] M M Galloway; Texture analysis using gray level run lengths, Computer Graphics and Image Processing, 4: 172-179, 1975

[Ge2005] Z Ge, B Sahiner, HP Chan; Computer-aided detection of lung nodules: false positive reduction using a 3D gradient field method and 3D ellipsoid fitting, *Medical Physics*, 32(8): 2443-54, 2005

[Gige94] M L Giger, K T Bae, H Macmahon; Computerized Detection Of Pulmonary Nodules In Computed-Tomography Images, *Investigative Radiology*, 29(4): 459-465, 1994

[Gurc01] M N Gurcan, N Petrick, B Sahiner, et al.; Computerized lung nodule detection on thoracic CT images: combined rule-based and statistical classifier for false-positive reduction, *Proc. SPIE, Medical Imaging,Image Processing*, v.4322: 686-692, 2001

[Gurc02] M N Gurcan, B Sahiner, N Petrick; Lung nodule detection on thoracic computed tomography images: preliminary evaluation of a computer-aided diagnosis system, Medical Physics, 29(11): 2552-2558, 2002

[Hagan95] M T Hagan, H B Demuth, M Beale; Neural Network Design, *PWS Publishing*, 1995

[Halif00] R Halif and J Flusser; Numerically Stable Direct Least Squares Fitting of Ellipses, Department of Software Engineering, Charles University, Czech Republic, 2000

[Hans00] D M Hansell; Imaging the lungs with computed tomography, *IEEE Engineering in Medicine and Biology Magazine*, 19(5): 71 – 79, 2000

[Hansel] D C Hanselman; University of Maine, Orono, ME 04469, Mastering MATLAB 7, 2005-02-28

[Haralic73] R M Haralick, K Shanmugam, I Dinstein; Textural features for image classification, IEEE Tran. On Systems, Man, And Cybernetics, Smc-3(6): 610-621, 1973

[Haralic79] R M Haralick; Statistical and structural approaches to texture, *Proceedings of the IEEE*, 67(5): 786 - 804, May 1979

[Hu2001] S Hu; E A Hoffman, J M Reinhardt; Automatic lung segmentation for accurate quantitation of volumetric X-ray CT images, *IEEE Trans. Medical Imaging*, 20(6): 490–498, 2001

[Hu2004] H Hu, P M. Holman; G de Haan; Image interpolation using classification-based neural networks, *IEEE International Symposium on Consumer Electronics*, pp. 133 – 137, 2004

[Huang96] L K Huang, M J Wang; Efficient shape matching through model-based shape recognition, *Pattern Recognition*, 29(2), 207-215, 1996

[IEEE90] IEEE standard glossary of image processing and pattern recognition terminology, 26 March 1990

[Jain96] A.K. Jain, J. Mao, K.M. Mohiuddin; Artificial Neural Networks: A Tutorial, (*IEEE*) *Computer*, 29(3): 31 – 44, 1996

[Jesa04] J P Jesan; The Neural Approach to Pattern Recognition, *Ubiquity*, 5(7), 2004

[Kanaza96] K Kanazawa, M Kubo, N Niki; Computer Aided Screening System for Lung Cancer Based on Helical CT Images, *Lecture Notes In Computer Science, Proceedings of the 4th International Conference on Visualization in Biomedical Computing*, 1131: 223 - 228, 1996

[Kane96] M Kaneko, K Eguchi, H Ohmatsu, et al.; Peripheral lung cancer: Screening and detection with low-dose spiral CT versus radiography, *Radiology* 201(3): 798-802, 1996

[Kawa00] Y Kawata, N Niki, H Omatsu et al.; Quantitative analysis of internal texture for classification of pulmonary nodules in three-dimensional thoracic images, *Proceedings of SPIE*, *Medical Imaging Image Processing*, v.3979: 863-871, 2000

[Kawa97] Y Kawata, N Niki, H Ohmatsu, et al.; Classification of pulmonary nodules in thin-section CT images based on shape characterization, *IEEE International Conference on Image Processing ICIP*, 3: 528-530, 1997

[Kawa99] Y Kawata, N Niki, H Ohmatsu, et al.; Classification of pulmonary nodules in thin-section CT images by using multi-scale curvature indexes, *IEEE International Conference on Image Processing ICIP*, v.2: 197 – 201, 1999

[Kim003] D Y Kim, J H Kim, SM Noh, et al.; Pulmonary nodule detection using chest CT images, *Acta Radiologica*, 44(3): 252-7, 2003

[Ko2001] J P Ko, M Betke; Chest CT: automated nodule detection and assessment of change over time—preliminary experience, *Radiology*, 218:267-273, 2001

[Ko2004] JP Ko, DP Naidich; Computer-Aided Diagnosis and The Evaluation of Lung Disease, *Journal Of Thoracic Imaging*, 19(3): 136-155, 2004

[Kosti03] W J Kostis, A P Reeves, D F Yankelevitz, et al.; Three-dimensional segmentation and growth-rate estimation of small pulmonary nodules in helical CT images, *IEEE Transactions on Medical Imaging*, 22(10): 1259-1274, 2003

[Kraemer] H C Kraemer, K K Lowe, and D J Kupfer. How to understand what research tells us about risk, *Oxford University Press*, New York, 2005

[Krup04] EA. Krupinski; Computer-aided Detection in Clinical Environment: Benefits and Challenges for Radiologists, *Radiology* 231(1): 7-9, 2004

[Kuhn06] JM. Kuhnigk, V. Dicken; L. Bornemann, et al.; Morphological segmentation and partial volume analysis for volumetry of solid pulmonary lesions in thoracic CT scans, *IEEE Transactions on Medical Imaging*, 25(4): 417 – 434, 2006

[Kung93] S Y Kung; Digital Neural Networks, *Prentice Hall*, 1993

[LakePET] Lake P.E.T. Imaging Center , Baton Rouge, LA, USA
http://www.ololrmc.com/body.cfm?xyzpdqabc=0&id=11&action=list

[Lasko05] T A Lasko, J G Bhagwat, K H Zou, and L Ohno-Machado; The use of receiver operating characteristic curves in biomedical informatics, *Journal of Biomedical Informatics*, 38: 404–415, 2005

[Lee001] Y Lee; T Hara.; H Fujita; Automated detection of pulmonary nodules in helical CT images based on an improved template-matching technique, *IEEE Trans. on Medical Imaging*, 20: 595-604, 2001

[Leon98] C L Leondes; Algorithms and Architectures, Neural Network Systems Techniques and applications, *Academic press*, v.1, 1998

[LeonC98] C L Leondes; Image processing and pattern recognition, *Neural Network Systems Techniques and applications*, v.5, Academic press, 1998

[Lewis03] M Lewis-Beck, A Bryman, T Futing (Eds.); Neural Networks, *In Encyclopedia of Social Sciences Research Methods*, 2003

[Li2003] Q Li, F Li, J Shiraishi, S Katsuragawa, et al.; Investigation of new psychophysical measures for evaluation of similar images on thoracic computed tomography for distinction between benign and malignant nodules, *Medical Physics*, 30(10): 2584–2593, 2003

[Lindeb91] T Lindeberg; Discrete Scale-Space Theory and the Scale-Space Primal Sketch, *PhD thesis*, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden, May 1991

[Looney97] C G Looney; Pattern Recognition Using Neural Networks, theory and algorithms for engineers and scientists, *Oxford University Press*, New York, p. 5, 1997

[Lou99] S L Lou, C L Chang, K P Lin, and T S Chen; Object-based deformation technique for 3-D CT lung nodule detection, *Proc SPIE*, 3661:1544-1552, 1999

[Maes90] M Maes; On A Cyclic String-To-String Correction Problem, *Information Processing Letters*, 35(2), 73-78, 1990

[Maes91] M Maes; Polygonal Shape Recognition Using String-Matching Techniques, *Pattern Recognition*, 24(5), 433-440, 1991

[McCul43] W S McCulloch, W Pitts; A logical calculus of the ideas imminent in neural activity, *Bull. Math. Biophys.*, v.5: 115–133, 1943

[McNitt99] M F McNitt-Gray, E M Hart, N Wyckoff, et al.; A pattern classification approach to characterizing solitary pulmonary nodules imaged on high resolution CT: Preliminary results, *Medical Physics*, 26(6): 880-888, 1999

[McWy99] M F McNitt-Gray, N Wyckoff, J WSayre, et al.; The effects of co-occurrence matrix based texture parameters on the classification of solitary pulmonary nodules imaged on computed tomography, *Computerized Medical Imaging and Graphics*, 23(6): 339-48, 1999

[Metz86] C E Metz; ROC methodology in radiographic imaging, *Invest Radiol*, 21:720–733, 1986

[NCIA] http://ncia.nci.nih.gov

[Nishida98] H Nishida; Matching And Recognition Of Deformed Closed Contours Based On Structural Transformation Models, *Pattern Recognition*, 31(10), 1557-1571, 1998

[Obuch03] N A Obuchowski; Receiver operating characteristic curves and their use in radiology, *RADIOLOGY*, 229(1): 3-8, OCT 2003

[Okumur98] T Okumura, T Miwa, JI Kako, et al.; Image processing for computer-aided diagnosis of lung cancer screening system by CT, *Proc. SPIE*, v.3338: 1314-1322, 1998

[Paik2004] D S Paik, C F Beaulieu, G D Rubin, et al.; Surface normal overlap: a computer-aided detection algorithm with application to colonic polyps and lung nodules in helical CT, *IEEE Transactions on Medical Imaging*, 23(6): 661-675, 2004

[Partain05] C L Partain, HP Chan, J G Gelovani, et al.; Biomedical Imaging Research Opportunities Workshop II: Report and Recommendations, *Radiology*, 236(2): 389 – 403, 2005

[Rogers01] D F Rogers; *An introduction to Nurbs with historical perspective*, Morgan Kaufman Pub., p. 44, 2001

[Rubin00] GD Rubin; Data explosion: the challenge of multidetector-row CT, *European Journal Of Radiology* 36(2): 74-80, 2000

[Satoh99] H Satoh, Y Ukai, N Niki, et al.; Computer aided diagnosis system for lung cancer based on retrospective helical CT image, *Proc. SPIE Medical Imaging*, 3661: 1324-1335, 1999

[Schalk97] R S Schalkoff; *Artificial Neural Networks*, MIT Press, p. 10, 1997

[Schreiber90] I Schreiber and M Ben-Bassat; Polygonal Object Recognition, *IEEE Proc. 10th Int. Con. on Pattern Recognition*, 1: 852 – 859, 1990

[Sluimer06] I Sluimer, A Schilham , M Prokop , et al.; Computer analysis of computed tomography scans of the lung: a survey, *IEEE Transactions on Medical Imaging*, 25(4): 385 – 405, 2006

[Sonka98] M Sonka, M Hlavac, R Boyle; *Image Processing Analysis and Machine Vision*, PWS, 2nd ed., p. 90, 1998

[Soysal06] Ö M Soysal, B Gunturk, K L Matthews II, Image Retrieval Using Canonical Cyclic String Representation of Polygons, *IEEE International Conference on Image Processing*, pp.1493 – 1496, Oct. 2006.

[Soysal07] Ö M Soysal and J Chen; Matching Polygons Using Canonical String representation, *IPCV'07 in WORLDCOMP'07*, pp. 407-413, 2007.

[Soysal08] Ö M Soysal and J Chen; Proc. A new spectral feature for shape comparison, *the Int'l Conf. in Image Processing, Computer Vision, & Pattern Recognition, IPCV'08 in WORLDCOMP'08*, pp.23-27, 2008

[Suzuki03] K Suzuki, S G Armato, F Li, S Sone S, et al.; Massive training artificial neural network (MTANN) for reduction of false positives in computerized detection of lung nodules in low-dose computed tomography, Medical Physics, 30(7): 1602-1617, 2003

[Tan2003] B B Tan, K R Flaherty, E A Kazerooni, M D Iannettoni; The Solitary Pulmonary Nodule, *CHEST*, Supplement, 123(1), p89S, 8p, 2003

[Takiza02] H Takizawa, S Yamamoto, T Matsumoto; Recognition of lung nodules from x-ray ct images using 3d markov random field models, *16th International Conference on Pattern Recognition*, 1(11-15): 99 – 102, 2002

[Wang03] Z Wang, Z Chi, and D Feng; Shape based leaf image retrieval, *IEEE Proc.-Vision Image And Signal Processing*, 150(1), 34-43, 2003

[Way2006] T W Way, L M Hadjiiski, B Sahiner, et al.; Computer-aided diagnosis of pulmonary nodules on CT scans: Segmentation and classification using 3D active contours, *Medical Physics*, 33(7): 2323-2337, 2006

[Wei2002] GQ Wei, L Fan, J Qian; Automatic detection of nodules attached to vessels in lung CT by volume projection analysis, *Lecture Notes in Computer Science*, *Medical Image Computing and Computer-Assisted Intervention*, *Springer*, v. 2488: 746–752, 2002

[Wei2006] GQ Wei, J Qian, L Fan; Vessel-feeding pulmonary nodule detection by volume projection analysis, *http://www.patentstorm.us/patents/7020316-fulltext.html*, No. 10008119, US Patent Issued on March 28, 2006

[WikiCan] *http://en.wikipedia.org/wiki/Cancer*

[Wilucan] *http://en.wikipedia.org/wiki/Lung_cancer*

[Wilung] *http://en.wikipedia.org/wiki/Lung*

[Witumor] *http://en.wikipedia.org/wiki/Tumor*

[Wolter03] D Wolter; Shape: Representation & Recognition, *lecture notes*, 2003

[Worman04] D Wormanns, S Diederich; Characterization of small pulmonary nodules by CT, *European Radiology*, 14: 1380–1391, 2004

[Wu2001] W Wu; Two-dimensional object recognition through string matching, *Imaging Science Journal*, 49(4), 213-221, 2001

[Zhang00] G P Zhang; Neural networks for classification: a survey, *IEEE Transactions on Systems, Man and Cybernetics*, Part C, 30(4): 451 – 462, 2000

[Zhang03] D Zhang, G Lu; Evaluation of MPEG 7 shape descriptors against other shape descriptors, *Multimedia Systems*, 9(1): 15-30, 2003

[Zhang05] X Zhang, G McLennan, E A Hoffman, et al.; Automated Detection of Small-Size Pulmonary Nodules Based on Helical CT Images, *Lecture Notes in Computer Science*, *Processing in Medical Imaging: 19th International Conference*, v.3565: 664-76, 2005

[Ziou97] D Ziou, S Tabbone; Edge detection techniques—an overview, *International Journal of Pattern Recognition and Image Analysis*, 8(4): 537-559, 1997

# APPENDIX: PERMISSIONS

Reprint Permission Letter from IEEE is given below.

28 January 2009

Omer M. Soysal  
Research Associate 5  
The Highway Safety Research Group <http://hsrg.lsu.edu/>  
Department of Information Systems and Decision Sciences, Louisiana State University  
ISDS Research Lab 3535 Nicholson Ext. Room# 100G Baton Rouge, LA 70803S

SUBJECT: Ömer Muhammet Soysal, Bahadir Gunturk, Kenneth L. Matthews II; "Image Retrieval Using Canonical Cyclic String Representation Of Polygons", The International Conference on Image Processing (ICIP), October, 2006.

Dear Omer M. Soysal:

This is in response to your email of 19 January 2009, in which you have requested permission to reprint, in your upcoming thesis/dissertation, your above described IEEE copyrighted material. We are happy to grant this permission.

In regards to distributing paper copies of your thesis or placing it on the university's website, please be sure that the following copyright/credit notice appears in the credits with the appropriate details filled in:  
© [Year] IEEE. Reprinted, with permission, from (complete publication information).  
For the website credits, please add the following:

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the [university 's name]'s products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org.

By choosing to view this material, you agree to all provisions of the copyright laws protecting it.  
Please be advised that wherever a copyright notice from another organization is displayed beneath a figure, a photo, a videotape or a Powerpoint presentation, you must get permission from that organization, as IEEE would not be the copyright holder.

Finally, if applicable, University Microfilms, Inc./ProQuest may supply single copies of the dissertation.

In the future, please be aware that most universities find our email permission grants acceptable and we are certain that yours will concur.

Sincerely,

Jacqueline Hansson
IEEE Intellectual Property Rights

Reprint Permission Request to IEEE is given below.

To: Representative, IEEE Copyrights and Permissions Section
Phone: +1 732 562 3966
Fax: +1 732 562 1746
copyrights@ieee.org
From: Omer Soysal
Highway Safety Research Group
Department of ISDS
Louisiana State University
omsoysal@lsu.edu; (225)-333-8934
Date: January 19, 2009
Re: Reprinting written permission letter of the article

Dear Editor;

Per regulation of the Graduate School at Louisiana State University, I like to kindly request a reprinting written permission letter of the article which was published by me and co-authors listed below as I like to use them in my dissertation.

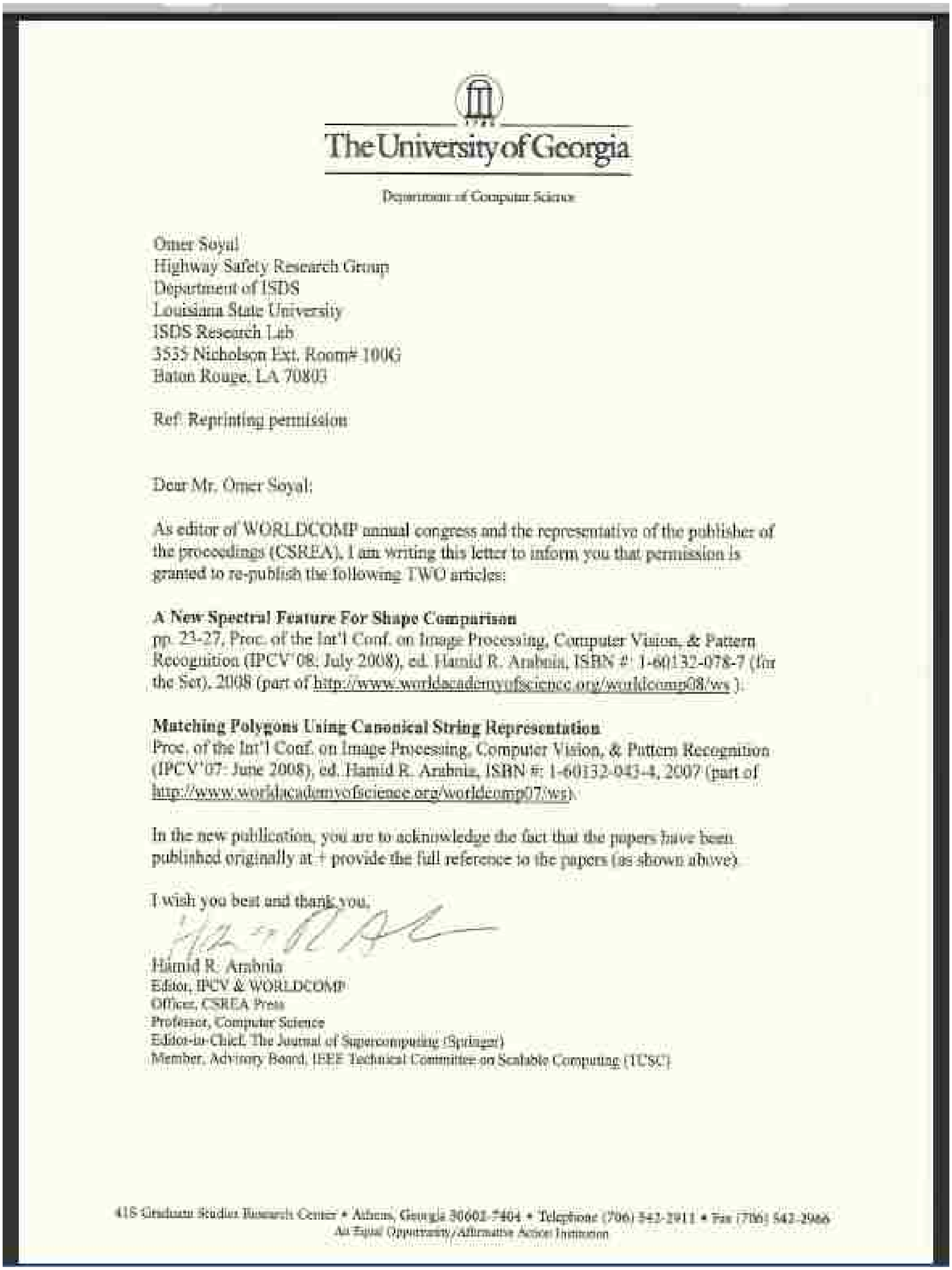
Please send the letter either via mail or e-mail. If e-mail is preferred, please send the letter as a pdf or Ms-Word file.

Regards,

The Paper:

Ömer Muhammet Soysal, Bahadir Gunturk, Kenneth L. Matthews II; "Image Retrieval Using Canonical Cyclic String Representation Of Polygons", The International Conference on Image Processing (ICIP), IEEE Signal Processing Society, October, 2006.

Reprint Permission Letter from IPCV is given below.



The University of Georgia

Department of Computer Science

Omer Soyal
Highway Safety Research Group
Department of ISDS
Louisiana State University
ISDS Research Lab
3535 Nicholson Ext. Room# 100G
Baton Rouge, LA 70803

Ref: Reprinting permission

Dear Mr. Omer Soyal:

As editor of WORLDCOMP annual congress and the representative of the publisher of the proceedings (CSREA), I am writing this letter to inform you that permission is granted to re-publish the following TWO articles:

**A New Spectral Feature For Shape Comparison**
pp. 23-27, Proc. of the Int'l Conf. on Image Processing, Computer Vision, & Pattern Recognition (IPCV'08: July 2008), ed. Hamid R. Arabnia, ISBN #: 1-60132-078-7 (for the Set), 2008 (part of http://www.worldacademyofscience.org/worldcomp08/ws ).

**Matching Polygons Using Canonical String Representation**
Proc. of the Int'l Conf. on Image Processing, Computer Vision, & Pattern Recognition (IPCV'07: June 2008), ed. Hamid R. Arabnia, ISBN #: 1-60132-043-4, 2007 (part of http://www.worldacademyofscience.org/worldcomp07/ws).

In the new publication, you are to acknowledge the fact that the papers have been published originally at + provide the full reference to the papers (as shown above).

I wish you best and thank you,

Hamid R. Arabnia
Editor, IPCV & WORLDCOMP
Officer, CSREA Press
Professor, Computer Science
Editor-in-Chief, The Journal of Supercomputing (Springer)
Member, Advisory Board, IEEE Technical Committee on Scalable Computing (TCSC)

415 Graduate Studies Research Center • Athens, Georgia 30602-7404 • Telephone (706) 542-2911 • Fax (706) 542-2966
An Equal Opportunity/Affirmative Action Institution

Reprint Permission Request to IPCV is given below.

To: Dr. Hamid R. Arabnia, Editor-in-chief of *WORLDCOMP*’07 & 08  
Department of Computer Science  
415 Graduate Studies Research Center  
Athens, Georgia 30602-7404, USA

From: Omer Soysal  
Highway Safety Research Group  
Department of ISDS  
Louisiana State University  
omsoysal@lsu.edu; (225)-333-8934

Date: January 19, 2009

Re: Reprinting written permission letter of the articles

Dear Editor;

Per regulation of the Graduate School at Louisiana State University, I like to kindly request a reprinting written permission letter of the articles which were published by me and co-authors listed below as I like to use them in my dissertation.

Please send the letter either via mail or e-mail. If e-mail is preferred, please send the letter as a pdf or Ms-Word file.

Regards,

List of the Papers:

Omer M Soysal, Jianhua Chen and Helmut Schneider. “A New Spectral Feature For Shape Comparison”. *IPCV’08 in WORLDCOMP’08*, pp. 23-27, 2008.

Omer M Soysal and Jianhua Chen. "Matching Polygons Using Canonical String Representation". IPCV’07 in WORLDCOMP’07, pp. 407-413, 2007.

# VITA

Ömer Muhammet Soysal was born in Giresun, Türkiye. He completed his Bachelor of Science degree at Hacettepe University, Ankara, Turkiye, in 1992. He earned his Master of Science degree in electronics at İnönü University, Malatya, Turkiye in 1998 and in computer science at Southern University, Baton Rouge, Louisiana, USA in 2002. He earned his doctorate in computer science at Louisiana State University in 2009.