EARLY DETECTION OF BREAST CANCER USING SUPPORT VECTOR MACHINE

**A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED SCIENCES
OF
NEAR EAST UNIVERSITY**

by

HÜSEYİN GÜNEY

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE**

**IN
COMPUTER ENGINEERING**

NICOSIA 2013

(Sırt)

H.GÜNEY

NEU, 2013

EARLY DETECTION OF BREAST CANCER USING SUPPORT VECTOR MACHINE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED SCIENCES
OF
NEAR EAST UNIVERSITY

by

HÜSEYİN GÜNEY

In Partial Fulfillment of the Requirements for
the Degree of Master of Science
in
Computer Engineering

NICOSIA 2013

Hüseyin Güney : Early Detection of Breast Cancer using Support Vector Machines

Approval of the Graduate School of Applied Sciences

Prof. Dr. İlkay Salihoğlu
Director

We certify this thesis is satisfactory for the award of the Degree of Master of Science in Computer Engineering

Examining Committee in charge:

Assist.Prof. Dr. Kaan Uyar, Committee Chairman, Computer Engineering Department, NEU

Assist.Prof. Dr. İbrahim Erşan, Committee Member, Computer Engineering Department, GAU

Assist.Prof. Dr.Firudin Muradov, Committee Member, Computer Engineering Department, NEU

Assist.Prof.Dr. Boran Şekeroğlu, Cosupervisor, Committee Member, Computer Engineering Department, NEU

Prof. Dr. Rahib H.Abiyev, Supervisor, Committee Member, Chairman of Computer Engineering Department, NEU

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Huseyin Guney

Signature :

Date:

ABSTRACT

Image classification is an attempt to label an image with appropriate identifiers. These identifiers are determined by the area of interest. Image classification is the process of assigning all pixels in a digital image to particular classes according to their characteristics. It is essential to extract the features of the images efficiently, without losing important color information, and reduce redundant color information. This can be done in two main approaches of image classification: supervised and unsupervised image classification. In the thesis the methodologies used in image classification are described briefly. It was shown that one of efficient methods used for image classification is supervised classification in particularly Support Vector Machine (SVM). In the thesis the medical image classification using support vector machine is presented. The implementation of software for early detection of breast cancer using SVM is performed. The main steps of breast cancer's image classification are explained. The main steps of image recognition based on image acquisition, image enhancement, feature selection and extraction and classification steps are performed. It was proved that SVM is very accurate and effective for the classification issues and it can be adjustable depending on the classes by using kernel functions. For this reason SVM is used for classification process of medical images. Implemented application and the results of the training and test processes show that early detection of breast cancer can be done in an accurate and efficient way. As a result, this thesis describes details of image classification

Key words: image classification, support vector machines, SVM, breast cancer, image acquisition, image enhancement, feature selection, feature extraction, classification techniques.

Eşime, Anneme ve Babama

To my Wife and my Parents

ACKNOWLEDGEMENTS

First and foremost I would like to thank my supervisor Prof. Dr. Rahib ABIYEV and who has shown plenty of encouragement, patience, and support as he guided me through this endeavor fostering my development as a graduate student and scientist. In addition, I would like to thank my co-supervisor Assist. Prof. Dr. Boran ŞEKEROĞLU for his important ideas and helps that assisted and help to improved my work. I am also thankful for the contributions and comments the teaching staff of the Department of Computer Engineering. Here also I would like to thank to my friends at the Department of Computer Engineering who helped me one way or the other.

This research was generously supported by the Department of Computer Engineering of the Near East University. I am grateful to all supporters.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Overview on Image Classification

Image classification is an attempt to label (often textual) an image with appropriate identifiers. These identifiers are determined by the area of interest, whether it is general classification for arbitrary pictures (for instance, from the internet), or a specific domain, for instance, medical x-ray images or geographical images of terrain. Image classification is the process of assigning all pixels in a digital image to particular classes according to their characteristics. This characterized data may then be used to produce thematic maps of the image itself. The objective of image classification is to identify and portray, as a unique grey level (or color), the features occurring in an image.

Nowadays computer aided (CAD) image classification systems are widely used in the different research areas like medical diagnosis, remote sensing, image analysis, pattern recognition etc. Image classification can be described as the process of sorting the important features of images into the classes. In additions, medical image classification CAD systems need to have high accuracy and efficiency while classification process because results of systems can be lead to physicians make wrong decision and incorrect treatments for diseases. As a result, computer in contrast to physicians can make more precise decisions with respect to the accuracy of developed system.

On the other hand, to make medical image classification systems more accurate, some sort of techniques have to be used. The most powerful and useful ones for this task are Gaussian smoothing filter, contrast stretching, Top Hat filtering, Binary conversion, Wavelet Transform, which are used for image enhancement and Support Vector Machine is used for classification.

For accurate image classification it is essential to extract the features of the images efficiently, without losing important color information, and reduce redundant color information. This can be done in two main approaches of image classification: supervised and unsupervised image classification.

Unsupervised image classification does not rely on a training set. Instead, it uses *clustering* techniques which measure the distance between images, and groups the images with common features together. This group can then be labeled with different class-identifiers. Unsupervised classification can be defined as the identification of natural groups or structures within the data. It clusters pixels in a data set based only on their statistics, without using previous knowledge about the spectral classes present in the image. Some of the more commonly used unsupervised classification methods are: Isodata (Witten & Frank, 2005) and k-Means (Witten & Frank, 2005). Moreover, unsupervised classification is a method which examines a large number of unknown pixels and divides them into a number of classes based on natural groupings present in the image values.

Unlike supervised classification, unsupervised classification does not require analyst-specified training data. The basic premise is that values within a given color pixel should be close together in the measurement space (i.e. have similar grey levels), whereas data in different classes should be comparatively well separated (i.e. have very different grey levels) (Lillesand & Kiefer, 1994). Besides that, supervised classification uses *training sets* of images to create descriptors for each class. The training sets are carefully manually selected to represent a common picture set of that class. The classifier method then analyses the training set, generating a descriptor for that particular class based on the common features of the training set. This descriptor could then be used on other images, which determines if that image is a part of that class. Supervised image classification is a subset of *supervised learning*. Supervised learning can generate models of two types. Most commonly, supervised learning generates a global model that inputs objects to desired outputs. In some cases, however, the map is implemented a set of local models. These local models are treated as inputs in such algorithms. Such algorithms are often implemented using *neural networks*, *decisions trees*, *support vector machines* and *Bayesian statistical methods*. The support vector machines show a great promise in this area.

1.2 Aim of the Thesis

The importance of this project is designing medical image classification system that can filter and locate a tumor area at a grayscale mammographic image of breast cancer. Generally, tumor area of breast is illustrated in the form of dense region in mammographic images. Therefore, by using image enhancement techniques tumor area of breast can be segmented and by using machine learning algorithm like SVM, normal and abnormal breast can be classified in an accurate and efficient way.

The main task to be accomplished in this project is implementing an image classification system using SVM for early detection of breast cancer and classification purposes. SVM is used for classification because it is very compromising machine learning system and has a good accuracy in results and it can be modified for linear and non-linear classification processes. To sum up, Mias breast cancer mammographic image database, image enhancement techniques, SVM and Matlab used to develop this application. In addition, the dataset is separated into train and test sections to develop and test the results of the application. Therefore, this application can be process images and classify them depending on the type of breast. It shows all images in a hyperplane and separate them by labeling them normal and tumor images.

1.3 Thesis Overview

The remaining chapters of this thesis are organized as follows:

- Chapter 2 introduces the types of image classification their advantages and disadvantages. Supervised and Unsupervised image classification techniques have been described. The importance of medical image classification and their practical implementation have been discussed.

- Chapter 3 describes the techniques used for medical image classification. Steps of image classification have been given. Medical image acquisition using Magnetic Resonance Imaging (MRI), Computer Tomography (CT), Mammography are explained. Image enhancement using stretching, filtering, wavelet transform are described. Also feature extraction and classification steps are presented.

- Chapter 4 presents the mathematical description of the support vector machines (SVM). Linear and nonlinear SVM, the used kernel functions are described. The importance of SVM in Image classification has been shown.

- Chapter 5 presents the development of medical image classification system using SVM. Image acquisition, enhancement, feature extraction and classification blocks are presented on breast cancer images. Classification simulations and results are listed at this section, accuracy of developed system listed and calculated. The implementation of image classification system has been done using Matlab package.

- Chapter 6 presents important results obtained from the thesis.

CHAPTER 2

REVIEW OF IMAGE CLASSIFICATION

2.1. Overview

In this section, brief information about image classification will be given and the used methodologies for image classification will be described. Basically medical image classification will be explained. Importance of the elements of artificial intelligence in image classification will be presented. Real world applications about image classification will be mentioned.

2.1 Review of Image Classification

Image classification is the one of important topics in the field of computer vision. Image classification plays an important role in areas of Medical diagnosis, Remote Sensing, Image analysis and Pattern Recognition. Digital image classification is the operation of sorting of images into a finite number of individual classes. Graphical representation of classification is given in Figure 2.1. Here the data describing the image is classified into two classes. In medical diagnosis, images have to be classified with maximum accuracy and efficiency. For instance, diagnosing of cells that have tumor is the one of most important task in medical image analysis. Nowadays the development of accurate image classification system for finding and classification tumors are become one of important problem in image processing. Therefore, image classification system can help humans to achieve their daily tasks. Otherwise, it will lead to incomplete treatment of the corresponding disease.

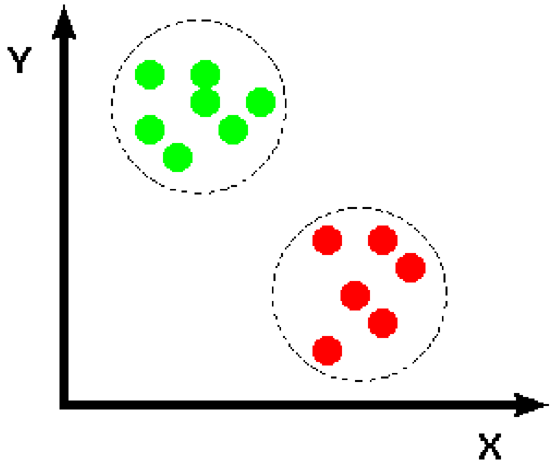


Figure 2.1 Graphical Representation of Classification. (Fisher, R., et. al., 2003, ¶ 1).

Image classification contains range of techniques to classify images depending on fields of images were taken. All algorithms that developed for image classification assumes every image has at least one feature, like spectral region of a land at remote sensing system, region of tumor area of a medical image, and each of these features belongs one or more classes. In addition, those classes can be specified by analyze of images which is supervised classification or automatically clustered of images which is unsupervised classification (Fisher, R., et. al. 2003). In other words, image classification uses information that contains digital number representation of images and tries to separate and classify each individual pixel of image depending on needed information. The aim of this system is assigning all related pixels to particular classes such as, water and forest in landscapes. In addition, the resulting classified image is a combination of pixels and it is a "thematic map" of the original image (Natural Resources Canada, 2008, ¶ 1).

The main idea is image classification system automatically categorize all pixels in an image into classes. In another word, it converts image data into information. There are two kinds of classes which are **information classes** and **spectral classes**. Information classes tries to define and separate particular parts in the image, such as different forest types or tree species, different geologic units or rock types, etc. Spectral classes form the group of similar pixels depending on their values like brightness in the different spectral channels of the data. The aim of image classification system while creating those classes is matching spectral classes in the data according to the interested region of information classes. Sometimes, there is a one-to-one match for those two classes. However, generally, those two groups do not match exactly. Using the forest example, spectral sub-classes may be due to variations in age, species, and density, or perhaps as a result of shadowing or variations in scene illumination. It is the analyst's job to decide on the utility of the different spectral classes and their correspondence to useful information classes (Natural Resources Canada, 2008, ¶ 2).

Finally, to sum up, image classification is the very important section of computer vision and artificial intelligence fields. It classifies images depending on analyzed data and it defines classes while doing this process. It creates information and spectral classes and matches them to classify images. Therefore, image classification plays an important role in areas of Medical diagnosis, Remote Sensing, Image analysis and Pattern Recognition.

2.2 Procedures of Image Classification

According to Gong and Howarth 1990, there are six steps of image classification procedures. Those procedures are listed below.

- Design image classification scheme: they are generally information classes such as urban, agriculture, forest areas, etc. Search studies in fields and gather base information and other ancillary data of the study area.
- Preprocessing of the image, including radiometric, atmospheric, geometric and topographic corrections, image enhancement, and initial image clustering.
- Select representative areas on the image and analyze the initial clustering results or generate training signatures.
- Image classification
    - Supervised mode: using training signature
    - Unsupervised mode: image clustering and cluster grouping
- Post-processing: complete geometric correction & filtering and classification decorating.
- Accuracy assessment: compare classification results with field studies.

2.3 How Image Classification Works

According to R. Fisher, et. al., image classification analyzes the numerical properties of various image features and organizes data into categories. Classification algorithms typically employ two phases of processing: training and testing. At the first stage of training process, attributes of typical image features are separated from each other. Depending on these, a unique description of each classification category is created and this is called a **training class**. In the following testing process, these feature-space partitions are used to classify image features (Fisher, R., et. al., 2003, ¶ 2).

Training classes are the important aspect of classification process. In supervised classification, statistical or distribution-free processes can be used to describe of classes. In unsupervised classification, classification process runs on clustering algorithms to do automatic segmentation of the training data into prototype classes. In addition, at both cases, there are some criteria for constructing training classes. They are:

- Independent: a change in the description of one training class should not change the value of another.
- Discriminatory: different image features should have significantly different descriptions.
- Reliable: all image features within a training group should share the common definitive descriptions of that group (Fisher, R., et. al., 2003, ¶ 2).

A reliable way of constructing a parametric definition of this sort is via a feature vector (V_1,V_2 … Vn) where n is the number of attributes which describe each image feature and training class. This representation allows us to consider each image feature as occupying a point, and each training class as occupying a sub-space (i.e. a representative point surrounded by some spread, or deviation), within the n-dimensional classification space. Viewed as such, the classification problem is that of determining to which sub-space class each feature vector belongs.

For instance, consider an application that distinguishes two different types of objects (e.g. bolts and sewing needles) based upon a set of two attribute classes (e.g. length along the major axis and head diameter). If it assumes that a vision system had the capability of extracting these features from a set of training images, result of this process can be shown below in the 2D feature space (Figure 2.2).

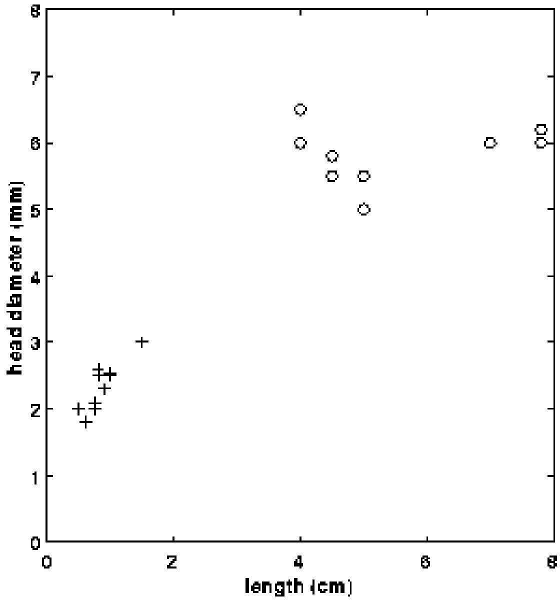


Figure 2.2. Graph of Feature Space: + sewing needles, o bolts. (Natural Resources Canada, 2008, ¶ 2).

At this point, important task is to define how to numerically partition the feature space so that the feature vector are used as a test object, we can determine, quantitatively, to which of the two classes it belongs. One of the simplest techniques is to employ a supervised, distribution-free approach known as the minimum (mean) distance classifier.

2.4 Types of Image Classification

The classification could be **supervised classification, unsupervised classification, textural classification, fuzzy classification,** etc. Those types can be used in different tasks depending on the aim of tasks because every type has its own strengths and weaknesses. However, supervised and unsupervised classification techniques are the most widely used ones because lots of algorithms are using those techniques, such as support vector machines (SVM), Artificial Neural Networks (ANN), etc. (Digital Image Processing, 2006).

2.4.1 Supervised Image Classification

In Figure 2.3 the steps of supervised image classification illustrated. Supervised image classification method need the image analyst to define the classes and let the system to do other steps that is assignment of pixels to the classes. In addition, supervised image classification method uses known pixels, which are defined by the image analyst of the system, to identify pixels of unknown classes (Grass Tutorial, n.d.), (Image Classification II, 2007). In addition, the computer assigns all of the remaining pixels to one of the predefined classes depending on the similarities of the classes. By using supervised classification, user defines the examples of the Information classes of interest in the image (Khalil, R., 2009). These are **training sites.** Then, image processing software develops a statistical characterization of the reflectance for each information class. This process is called **signature analysis** and involves developing a characterization as simple as the mean or the rage of reflectance on each bands, or as complex as detailed analyses of the mean, variances and covariance over all bands. Once a statistical characterization has been achieved for each information class, the image is then classified by examining the reflectance for each pixel and making a decision about which of the signatures it resembles most (Tangjaitrong, S., 1999).

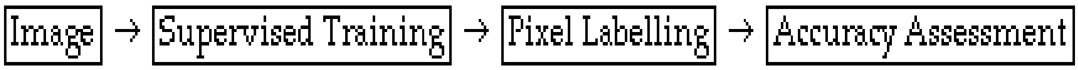


Figure 2.3. Steps of Supervised Image Classification (Gong, P., 1997, ¶ 1).

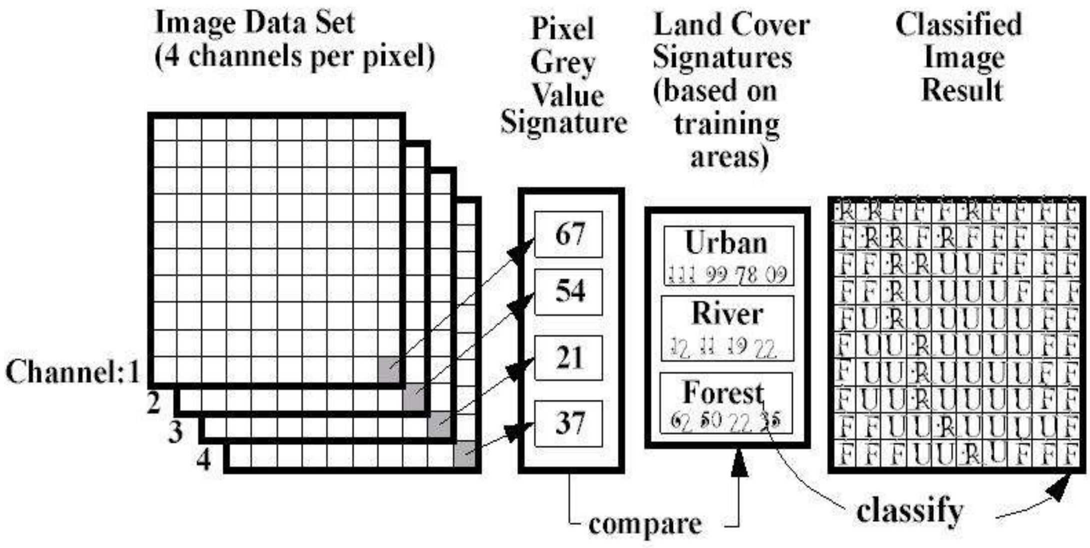


Figure 2.4. Steps of Image Classification (Tangjaitrong, S., 1999).

2.4.1.1 Advantages of Supervised Image Classification

The main advantages of the supervised image classification method may be listed as follow.

- Generates information classes
- Self-assessment using training sites
- Training sites are reusable

2.4.1.2 Disadvantages of Supervised Image Classification

The main disadvantages of the supervised image classification method may be listed at follow.

- Information classes may not match spectral classes
- Signature homogeneity of information classes varies
- Signature uniformity of a class may vary
- Difficulty and cost of selecting training sites
- Training sites may not encompass unique spectral classes

2.4.1.3 Procedures of Supervised Image Classification

There are five steps for procedures of supervised image classification. All steps of supervised image classification are listed below with detailed explanations.

- <u>Determines a classification scheme:</u> Classification scheme actually is equal to structure of classes. Classes of supervised classification system are typically created with a specific goal or target in mind. By defining classes' right, classification will be less ambiguities and inconsistent. However, not all data can be match into a "class" because of fuzzy or mixed areas within the image. There are often no clear boundaries in the image. Therefore, determining classification scheme is very important and it has to be correct for accurate classification (Bryant, R. G., 1999) & (McGinty, C., 2006).
- <u>Selects training sites on image:</u> Analyst requires selecting training sites based on the knowledge that gathered from the task and its images (McGinty, C., 2006).
- <u>Generates class signatures:</u> Training areas characterize spectral properties of classes and assigning other pixels to classes by matching with spectral properties of training sets (Bryant, R. G., 1999).

- Evaluates class signatures: Clusters are spectrally distinct and signatures are informationally distinct and when using the supervised procedure, the analyst must ensure that the informationally distinct signatures are spectrally distinct (Bryant, R. G., 1999).
- Assigns pixels to classes using a classifier: Using classification algorithms to classify parts in images (Image Classification II, 2007).

2.4.2 Unsupervised Image Classification

Steps of unsupervised image classification is illustrated in Figure 2.5.

Image → Clustering & Cluster Analysis → Clustering & Cluster Grouping → Accuracy Assessment

Figure 2.5. Steps of Unsupervised Image Classification (Gong, P., 1997, ¶ 1).

According to the Dr. Ragab Khalil, unsupervised image classification runs on the computer to classify spectrally-similar pixels into classes (Khalil, R., 2009). Furthermore, unsupervised classification is the identification of natural parties with multispectral data. Unsupervised classification does not use training data for information classes for the classification. In addition, pixels of images are evaluated and combined into several spectral classes depending on natural clustering in multi-dimensional space. Moreover, unsupervised classification is the definition, identification, labeling and mapping of natural spectral classes (Bryant, R. G., 1999). On the other hand, unsupervised classification process includes two groups' jobs that are analyst's job and computer's job. Computer's job is using algorithms to combine similar pixels into classes according to their similarities with each other and dissimilarities to other remaining pixels based on images. In addition, computer has no information about image areas and each class is initially unknown. Therefore, image analyst's job is to match the classes defined by the computer.

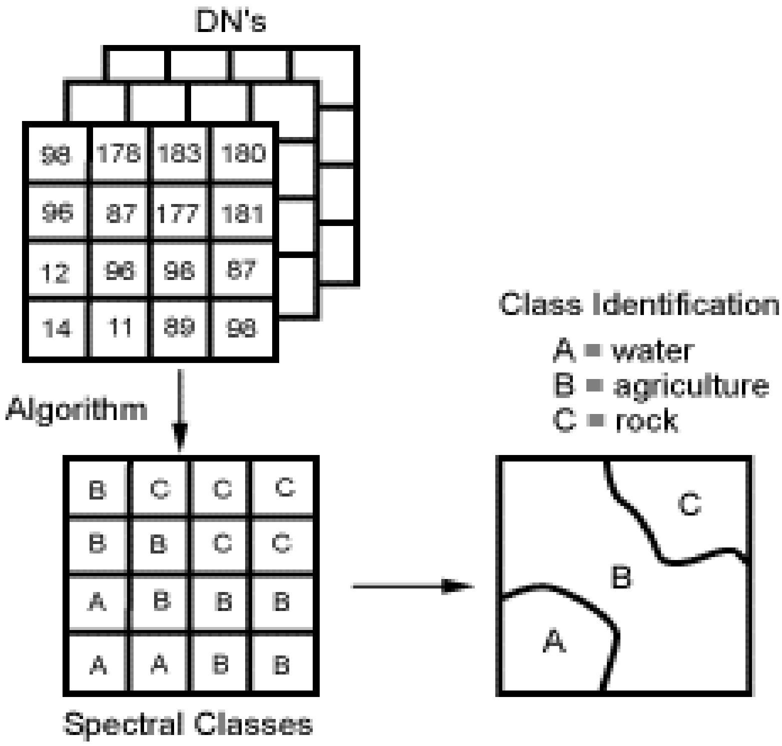


Figure 2.6. Spectral Classes Class Identification (Bryant, R. G., 1999).

2.4.2.1 Advantages of Unsupervised Image Classification

The main advantages of the unsupervised image classification method may be listed as follows.

- The computer system can match pixels to spectrally-distinct classes that an analyst may not recognize those (Image Classification Types, n. d.).
- The computer can specify a larger number of spectrally-distinct classes than an analyst can consider exist classes (Image Classification Types, n. d.).

2.4.2.2 Disadvantages of Unsupervised Image Classification

The main disadvantages of the unsupervised image classification method may be listed as follows.

- Spectral grouping might not be correspond to information classes because it generated by the classifier (Image Classification Types, n. d.).

- Spectral properties of particular classes may change over time depending on information and spectral classes because they are not constant classes (Image Classification Types, n. d.).

2.4.2.3 Procedures of Unsupervised Image Classification

- According to analyst's knowledge or user requirements, intervals of number of categories are generated by classification algorithms (Bryant, R. G., 1999).
- To form clusters and their centers, random selections of pixels are generated (Bryant, R. G., 1999).
- According to criteria's that defined by user, algorithm is picked to find distance between pixels and create starting values for estimation of cluster centers (Bryant, R. G., 1999).
- After adding pixels to initial estimations, means of new classes computed. This operation continues iteratively until the mean does not change from one of the iteration to another, significantly (Bryant, R. G., 1999).

2.4.3 Supervised vs. Unsupervised Image Classification

As mentioned above, there are two types of image classification, Supervised and Unsupervised classification. Both of them have their advantages and disadvantages in contrast to other. That is, both of them have their strengths and weaknesses. At this section, I would like to explain comparison of both supervised and unsupervised image classification to understand these principles better.

The main difference between supervised and unsupervised classification systems is supervised classification includes **prior decision process** and unsupervised classification includes **posterior decision process**. To define prior decision process of supervised image classification, we can say that image analyst supervises the selection of regions represent the features that analyst can recognize. On the other hand, to define posterior decision process of unsupervised classification, statistical clustering algorithms used to select classes formed by data. According to this processes, unsupervised classification is more based on computer automation. In addition, supervised prior decision works in flow from classes in the image to clusters in feature space and unsupervised posterior decision works in flow from clusters in feature space to classes in the image. Actually, both supervised and unsupervised image classification uses same components to classify images but they have some differences and the main one is supervised classification working from classes of image to cluster in feature space and unsupervised classification works in vice versa. Prior and posterior decision processes are illustrated at the figure below (Muhammad, H. H., 2006).

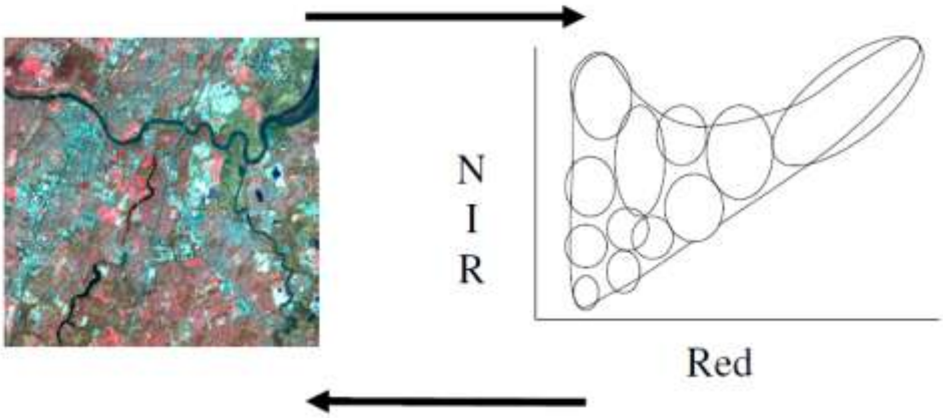


Figure 2.7. Supervised and Unsupervised Decision Process (Prior & Posterior Decision) (Muhammad, H. H., 2006, p. 4).

Finally, all the steps of supervised and unsupervised image classification are shown in the Figure 2.8.

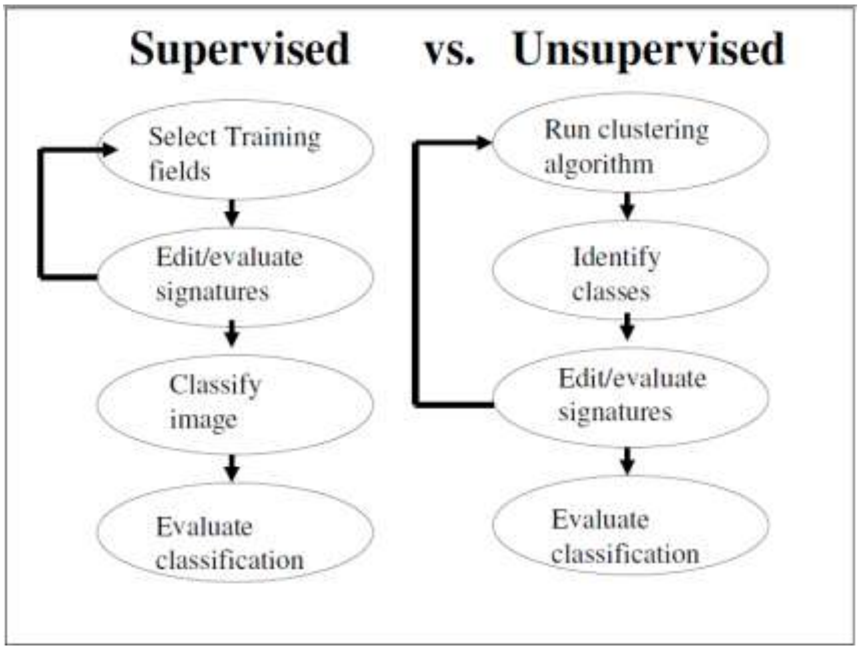


Figure 2.8. Supervised vs. Unsupervised Classification Algorithm and Chart (Muhammad, H. H., 2006, p. 3).

2.5 Practical Applications of Image Classification

Image classification is a principle to use for classification of images by using some techniques like supervised or unsupervised image classification. According to their pros and cons, they can be applied in different fields. Therefore, some of those fields are listed below.

- Medical Imaging: Medical image classification has an important statue in the field of medicine for disease diagnostic purposes. For these topics, different models of images are created and used. Therefore, there are many classification techniques with respect to grayscale and color medical images which acquired from medical devices (Smitha P., Shaji L., & Mini, Dr. M. G.,2011).
- Remote sensing Imaging – Locate objects in satellite images (roads, forests, etc.): According to (CRISP, Liew, Dr. S. C., 2001) website, “different land cover types in an image can be discriminated using some image classification algorithms using spectral features, i.e. the brightness and "color" information contained in each pixel. The classification procedures can be "supervised" or "unsupervised".”
- Image Pattern Recognition: Tries to classify images by generating their descriptions and relating them to their classes. (Rosenfeld, A., 2005).
- Agricultural Imaging – Crop Disease Detection: According to R. Kumor et. al., 2011, “The management of perennial fruit crops requires close monitoring especially for the management of diseases that can affect production significantly and subsequently the post-harvest life.”

  The image processing can be used in agricultural applications for following purposes:

  a. To detect diseased leaf, stem, fruit

  b. To quantify affected area by disease.

  c. To find shape of affected area.

  d. To determine color of affected area

  e. To determine size & shape of fruits etc.

  (Patih, J. K., & Kumor, R., 2011).

2.6 Medical Image Classification

First of all, medical image classification is the sub-discipline of the image classification subject. It is implemented and improved to classify abnormalities at the images of human body like tumor in woman’s breast, tumor in brain etc. Secondly, after some important improvements are done in medical field, medical imaging techniques have been started to use widely in medical field. According to medical imaging techniques, there are many different devices and images depending on those devices are implemented. Therefore, classification and processing of medical images became a must in the field. That is, medical imaging techniques generate lots of images that are including information about anatomical structures being examined and that information lead physicians and systems to make correct diagnoses, finding most suitable therapy, surveying phases of the treatment and so on (Dobrescu, R. et. al., 2010). At this point, in order to done these tasks in an automation and more accurate and efficient way, implementation of the medical image classification systems became compulsory. Therefore, medical image classification system improved to do these tasks depending on the medical images and type of the diseases. Finally, many types of classification techniques are created for medical image classification that can work on both grayscale and color images. There are mainly two ways to achieve medical image classification task which are texture classification and classification using machine learning like Artificial Neural Network, Support Vector Machines, etc. and classification using data mining techniques. Texture classification techniques try to find and locate different regions on images depending on the texture of the image. Then, it collects data and analyses textures to make classification process. Machine learning systems use supervised or unsupervised learning algorithms and try to understand differences in the image. In addition, they can improve themselves depending on the number of the images proceed. The last one, data mining techniques mean that analysis of large amount of data to find differences in the image (Smitha P. et. al, 2011).

CHAPTER 3

MEDICAL IMAGE CLASSIFICATION TECHNIQUES

3.1 Overview

At this section, medical image classification and appropriate techniques that are used to achieve image classification in an accurately and efficient way is mentioned. Those techniques can be categorized in steps that are followed while classifying images. The basic steps of image classification are "Image Acquisition", "Image Enhancement", "Feature Extraction", and "Classification" (Figure 3.1). In addition, all these steps have their own sub steps in them and each sub step is a technique to edit and correct images, to extract features and classify obtained data. Image acquisition step contains lots of techniques about obtaining images for classification purposes, such as MRI, Ultrasound, CT, Mammogram etc. medical devices are used to acquire images for medical purposes, satellites are used to acquire images for land use imaging purposes, digital cameras are used to acquire images for traffic control systems etc. To sum up, Image Acquisition step is used to gather images for classification purposes with respect to area of image classification will be achieved. Image Enhancement technique includes operations like image filtering, smoothing, edge detection, contrast stretching etc. Moreover, Feature Extraction techniques are made up of operations that contain features which are generated by using unique features of images. In addition, Classification technique uses methods that are used to classify images like Support Vector Machines (SVM), Artificial Neural Networks (ANN), etc. As a result, there are lots of techniques are used to classify images depending on each step and category.

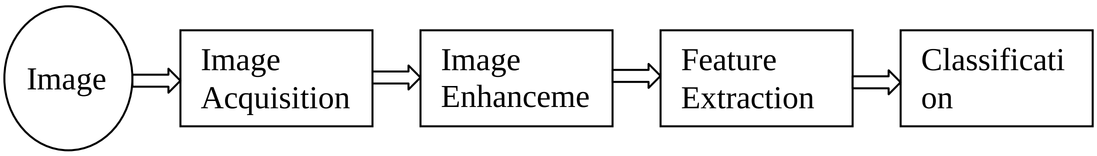


Figure 3.1 Steps of image classification

3.2 Image Acquisition

As mentioned above section, image acquisition is obtaining images from imaging devices or storage areas. For example, getting MRI images from MRI device or getting some images from digital cameras etc. In other words, image acquisition is the creation and gathering digital images from physical sources. It can be include processing, compression, storing and printing of those taken images.

Image acquisition step is a self explanatory and it does not need any more detailed explanations.

3.2.1 Medical Image Acquisition

In order to understand medical image acquisition, medical imaging and acquisition devices should be understood well, first. Therefore, at this section, first medical imaging and then acquisition devices in medical will be explained.

3.2.2 Medical Imaging

According to U.S. Food and Drug Administration website, “Medical imaging refers to several different technologies that are used to view the human body in order to diagnose, monitor, or treat medical conditions. Each type of technology gives different information about the area of the body being studied or treated, related to possible disease, injury, or the effectiveness of medical treatment (U.S. FDA, 2013).Furthermore, medical imaging is made of a subtitle biomedical engineering, medical physics or medicine with respect to the context that are research and development in the area of instrumentation, image acquisition (e.g. radiography), modeling and quantification are usually the preserve of biomedical engineering, medical physics, and computer science, and research into the application and interpretation of medical images is usually the preserve of radiology and the medical sub-discipline relevant to medical condition or area of medical science under investigation.

Many of the techniques that are developed for medical imaging also have scientific and industrial applications. Finally, medical imaging techniques create huge amount of data, especially CT, MRI and PET devices. Therefore, storage and communication of electronic image data is a problem and because of this problem, a compression technique has to be used. So that, JPEG 2000 image compression is used for DICOM standard for storage and transmission of medical images. In addition, JPIP standard is used to enable efficient streaming of the JPEG 2000 compressed images.

At the medical imaging issue, one of the most important topics is the medical imaging procedures that scientists acquire medical images using these machines and their imaging procedures. There are several medical imaging procedures depending on their aim and type.

3.2.2.1 Magnetic Resonance Imaging (MRI)

Magnetic Resonance Imaging (MRI) is an imaging process that creates and activates a strong magnetic field for magnetizing protons in the tissues of the human body and that magnetic field can be in the range from 1.5 Tesla to 3 Tesla. According to this process, Radio Frequency triggers protons and started energy absorption and re-emission of RF signals. After that, magnetic characteristics of tissues are detected and operate them in the form of grayscale images. In addition, sequences of pulses create differences in tissue contrast and that is the bases of different MRI studies. As a result, T1, T2, proton density, blood flow, perfusion, and diffusion are tissue characteristics used by MRI to change tissue contrast (Seibert, J. A., 2012).

Magnetic Resonance Imaging generates a two dimensional grayscale image of a thin slice of the human body and modern MRI instruments are also able to form three dimensional blocks of the human body. In addition, image sizes of MRI device may vary and they can be in a square and non-square matrix forms. For example, 64x64, 64x128, 128x128, 128x192, 256x512, and so on (Seibert, J. A., 2012).

3.2.2.2 Computer Tomography (CT)

According to Invert Website, "tomography is the process is which an object is viewed at multiple angles, and the results processed by a computer to calculate the object's internal structure." (Invert Website, 2012).

Computed Tomography (CT) scanners obtain data, which is thin-slice projection, by using a rotation of x-ray tube and detector array. After that, they generate images of anatomical volume using computer reconstruction algorithms. In addition, reconstructed image size for axial images is generally 512x512x12 bits (Seibert, J. A., 2012).

3.2.2.3 Mammography

According to the website of Radiological Society of North America, "Mammography is a specific type of imaging that uses a low-dose x-ray system to examine breasts. A mammography exam, called a mammogram, is used to aid in the early detection and diagnosis of breast diseases in women." (Radiological Society of North America Website, 2013).

On the other hand, digital mammography is the evolved version of traditional mammography because it uses digital receptors and computer systems in place of x-ray film to examine breast tissue for breast cancer. The electrical signals can be shown on computer screens and it allows evaluation of images to get more precise results. Digital mammography image size can be vary in the range from 0.1mm to 0.05mm detector element size and 12 or 16 bits of pixels to create image size of 8 to 50 MB (Seibert, J. A., 2012).

3.3 Image Enhancement

Image enhancement technique is an under discipline of computer graphics because this technique deals with computer graphics which are digitally stored in electronic media. Image enhancement could be defined as the process of improving the quality, making the image better depending on purposes of the user, by using some sort of imaging software. For example, making an image darker or lighter, increasing or decreasing the contrast of image, removing noises in the image, etc. In addition, image enhancement technique can be group into two sections that are simple and advanced image enhancement techniques. The simple image enhancement techniques include only operations like increasing contrast or making the image lighter. However, advanced image enhancement techniques contain operations like removing noises, smoothing filters etc. Moreover, in the computer field, there are lots of programs that are called as image editors that are created for image enhancement purposes. Some of them have capability to do advance image enhancement techniques and some of them have not.

Image enhancement is very important for medical imaging because disease diagnostic techniques in medicine use imaging technologies and those taken images needed to be manipulate to detect diseases. For example, by using image enhancement techniques, region of a tumor in brain of breast can be identified by computer systems. In addition, in medicine image enhancement techniques can be used for enhance contrast of local features, remove noise and other artifacts, enhance edges and boundaries, and composite multiple images for a more comprehensive view (Mueller, K., 2007).

Image enhancement has basic two operations which are local and global operations. Global operations operate on the whole set of pixels at once. For instance, brightness and contrast enhancement. Local operations operate on a set of pixels which are neighbor with each others. For example, edge detection, contouring, image sharpening, blurring.

According to the operation areas, local and global, image enhancement technique uses sort of methods to achieve these operations and that mentioned methods are Spatial and Frequency domain methods. Spatial domain method means that combination of pixels that are constructing an image and this method works on these pixels. In addition, Frequency domain method is the computation of Fourier Transform of an image, filtering result of Fourier Transform and taking the inverse of transform. According to these methods, there are techniques that are belong to mentioned methods above; Contrast Enhancement, Median/Max/Min Filtering, Gaussian Filtering, Top Hat Filtering, Image Subtraction, Histogram Equalization, Image Smoothing, Neighborhood Averaging, Transforms, Edge Detection and Image Sharpening. As a result, at this section, I have explained image enhancement techniques and sub-disciplines and at the following parts, I will describe those techniques (Image Enhancement Techniques, 2012).

3.3.1 Contrast Enhancement

Contrast enhancement technique is the one of subtitles of image enhancement technique. Briefly, contrast enhancement can be defined as the process that makes the light colors lighter and dark colors darker to increase the total contrast of an image simultaneously. Contrast enhancement technique uses following processes to achieve increasing contrast of any image. To achieve this operation, firstly, it specifies two boundaries which are lower and upper boundary. In addition, all color components in the image which are under the lower boundary rounded down to zero and above the upper boundary rounded up to possible maximum intense value (Gruber, T., 2001).

The aim of contrast enhancement technique is improving the contrast of the image depending on color differences in the image to changing the brightness difference between objects and their backgrounds. In addition, contrast enhancement is the process of contrast stretching and tonal enhancement in order or at one step. Contrast stretching improve the brightness differences uniformly but tonal enhancement improves the brightness differences in shadow (dark), midtone (gray) and highlight (bright) regions of an image (Fiete, R. D., 2010).

3.3.2 Contrast Stretching

The main idea of contrast stretching is increasing the dynamic range of gray levels in a grayscale image. Contrast stretching operates on the histogram values in the active layer of image. According the type of image, it finds minimum and maximum values for each channel and stretches all of them depending on the minimum and maximum values to make dark regions darker and light regions lighter. As a result, contrast stretching can be used for removing undesirable colors from an image which are pure white or pure black (The GIMP Documentation Team, 2012). An example of contrast stretching operation can be shown at the following photograph.

Figure 3.2. An example of Contrast Stretching Operation (Kolas, O., 2005).

3.3.3 Image Filtering

Acquired images can be corrupted or affected by random variations in intensity, illumination or had poor contrast. Because of this, they may not be used for some reasons and needed to be fixed. For achieving this problem, some sort of operations like image filtering can be applied. In addition, Image Filtering transforms pixel intensity values of images to make some sort of image characteristics visible and operable on them. While using image filtering techniques, enhancement can be used for improve contrast, smoothing for removing noises and template matching for detecting patterns (Petrakis, E. G. M., 2003).

3.3.3.1 Min and Max Filtering

Minimum and maximum filters are also called as erosion and dilation filters, respectively and these filtering techniques are belong to the morphological filters. Both filtering techniques operate on pixels on a specific area which are neighbor with each others. In addition, from the list of neighbor pixels, the minimum or maximum value is placed and stored as a resulting value. Therefore, resulting value that created for its related neighborhood is replaced by the resulting value in the each pixel of the image (Astrophoto, P., 2010).

Minimum filtering improves the dark places in the image by increasing its neighborhood area. It can be operated on any window size and is operated for the darkest surrounding pixel. Then, darkest pixel becomes the new value of the center of the selected window. For example, for the window (22 77 48, 150 77 158, 0 77 219) the center value could be changed from 77 to 0. After that operation, a resulting image can be generated as follow (RoboRealm, 2005).

Source

Min Filter

Figure 3.3. An example of Min Filter Operation (RoboRealm, 2005).

Maximum filtering improves the bright places in the image by increasing its neighborhood area. It can be operated on any window size and is operated for the brightest surrounding pixel. Then, brightest pixel becomes the new value of the center of the selected window. For example, for the window (22 77 48, 150 77 158, 0 77 219) the center value could be changed from 77 to 219. After that operation, a resulting image can be generated as follow (RoboRealm, 2005).

Source

Max Filter

Figure3.4. An example of Max Filter Operation (RoboRealm, 2005).

Both min and max filters work on neighborhood of pixels in image and make them darker or brighter by detecting the center and modifying with darkest or brightest value.

3.3.3.2 Mean and Median Filtering

Both mean and median filters can be applied to remove noise from an image. Mean filter takes the average of the current pixels and its neighbors, and Median filter makes same operation with Mean filter but it takes the median of current pixels and its neighbors. In addition, Median filter sorts all the values from low to high and takes the value in the center. However, if there are two values in the center, average of both is taken. At mean filter, it takes all the pixels and takes the average and put it in the center of current pixels. As a result, median pixel gives better results for salt and pepper noise because it completely removes noise. However, at mean filter, color of noise particles could be included to the average calculations and it affects the results of filter operation. Moreover, median filter reduces the image quality but mean filter do not (Vandevenne, L., 2004 & Schulze, M. A., 2001). At the following photos, examples of mean and median filters are shown.

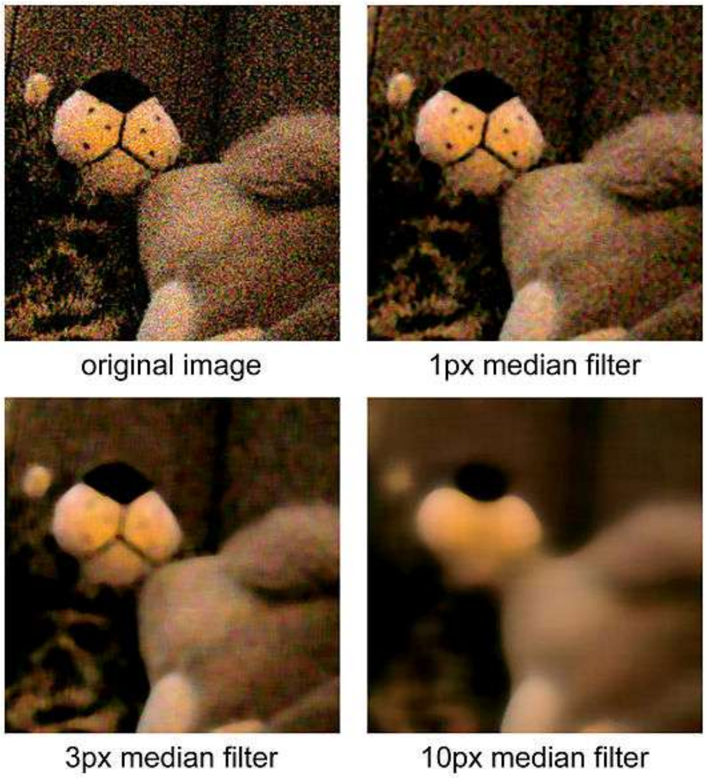


Figure 3.5 An example of Median Filter Operation (Wikipedia, 2012).

Figure 3.6 An example of Mean Filter Operation (Vandevenne, L., 2004).

3.3.3.3 Gaussian Smoothing Filtering

According to Computer Vision Demonstration Website of University of Southampton, "The Gaussian Smoothing Operator performs a weighted average of surrounding pixels based on the Gaussian distribution. It is used to remove Gaussian noise and is a realistic model of defocused lens". In addition, sigma defines the amount of blurring and the radius slider is used to control how large the template is. Large values for sigma will only give large blurring for larger template sizes (Nixon, M., & Aguado, A., 2002).

Figure 3.7 An example of Gaussian Smoothing Filter (Nixon, M., & Aguado, A., 2002).

Finally, Gaussian smoothing filter operator creates a template of values which contain group pixels and filter operation applied to them. Furthermore, values of these templates are defined by 2D Gaussian Equation which is illustrated at below (Nixon, M., & Aguado, A., 2002).

$$\frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{3.1}$$

3.3.3.4 Top-Hat Filtering

Top-hat filtering calculates the morphological opening of the image and then subtracts the results from the original image (Mathworks, 2012).

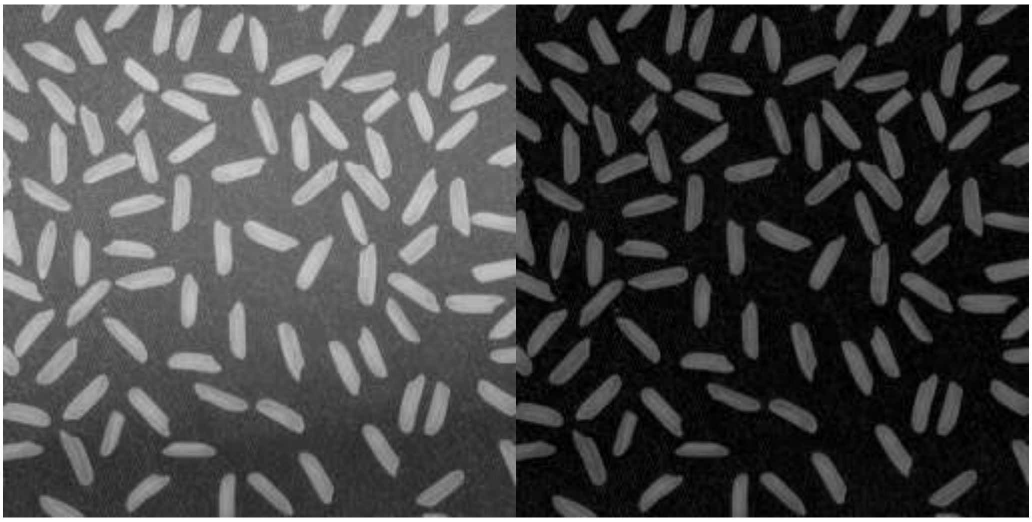


Figure 3.8 An example of Top-Hat Filter (Mathworks, 2012).

3.3.3.5 Image Transforms

Image transform may be used to transform an image from one domain to other one. Putting images in domains like frequency or Hough space can allow to identify of features which may not detected easily in the spatial domain. There are some subtitles for image transforms which are Radon Transform, Hough Transform, Discrete Cosine Transform, Discrete Fourier Transform and Wavelet Transform. All these transforms listed and explained below (Mathworks, 2012).

- **Radon transform**, used to reconstruct images from fan-beam and parallel-beam projection data
- **Hough transform**, used to find lines in an image
- **Discrete cosine transform**, used in image and video compression
- **Discrete Fourier transform**, used in filtering and frequency analysis
- **Wavelet transform**, used to perform discrete wavelet analysis, denoise, fuse images (Mathworks, 2012).

3.3.3.5.1 Wavelet Transforms

Wavelet transforms are mathematical averages to operate signal analysis when signal frequency can differentiate over time. In other words, the wavelet transform can determine frequency or scale components simultaneously with their location and time. In addition, calculates are directly proportional to the length of the input signal. Additionally, speech and audio processing, image and video processing, biomedical imaging and 1-D and 2-D applications in communications and geophysics are the applications of the wavelet transform. Finally, Wavelet transforms are in two distinct classes which are continuous and discrete wavelet transforms (Mathworks, 2012, Addison, P. S., 2005, & Bruce, A. et. al., 1996).

3.3.3.5.1.1 Continuous Wavelet Transforms

Continuous Wavelet Transform (CWT) uses inner product to calculate a signal and an analyzing function to find similarity between them. CWT compares the signal to shifted and compressed or stretched versions of a wavelet. Mathematical notation of CWT is shown as (Mathworks, 2012).

$$Xw\,(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} x(t)\,\psi\, * \left(\frac{t-b}{a}\right) dt \tag{3.2}$$

According to Matlab documentation website, in the CWT, the analyzing function is a wavelet, ψ. The CWT compares the signal to shifted and compressed or stretched versions of a wavelet. Stretching or compressing a function is collectively referred to as dilation or scaling and corresponds to the physical notion of scale. By comparing the signal to the wavelet at various scales and positions, you obtain a function of two variables. The two-dimensional representation of a one-dimensional signal is redundant. If the wavelet is complex-valued, the CWT is a complex-valued function of scale and position. If the signal is real-valued, the CWT is a real-valued function of scale and position. For a scale parameter, $a>0$, and position, b, the CWT (Mathworks, 2012) is:

$$C(a,b;\, f(t),\psi(t)) = \int_{-\infty}^{\infty} f(t)\, \frac{1}{\sqrt{a}}\, \psi\, * \left(\frac{t-b}{a}\right) dt \tag{3.3}$$

3.3.3.5.1.2 Discrete Wavelet Transforms

The discrete wavelet transform is the sub-title of wavelet transforms. It is constructed on sub-band coding technique which is created to do fast computation of wavelet transforms. Therefore, discrete wavelet transform can be implemented easily and it is required less time and resources for computations. In other words, DWT is an implementation of the wavelet transform depending on discrete set of the wavelet scales and translations in order to obey some defined rules (Klapetek, P., Necas, D., & Anderson, C., 2012).

According to Gwyddion website, “DWT decomposes the signal into mutually orthogonal set of wavelets, which is the main difference from the continuous wavelet transform. The wavelet can be constructed from a scaling function which describes its scaling properties” (Klapetek, P. et. al., 2012).

The mathematical formulation of the DWT is shown below. That formula is created for one level of transformation. However, it can be modified and it can be applied up to n level of transformation (Olkkonen H., 2011).

$$y[n] = \sum_{i=1} h_i x[n-k] \quad \text{for} \quad k = 1,2,3, \ldots, m\text{-}1 \tag{3.4}$$

DWT is providing solutions for signal processes with high accuracy. DWT also can provide accurate solutions for image processing at the aspects of compression and denoising. For example, JPEG2000 is created for image compression using wavelets. For image denoising, n-dimensional wavelet decomposition and reconstruction are used. As a result, DWT can be used in signal and image processing and the photo at the following shows how it works at image decomposition.

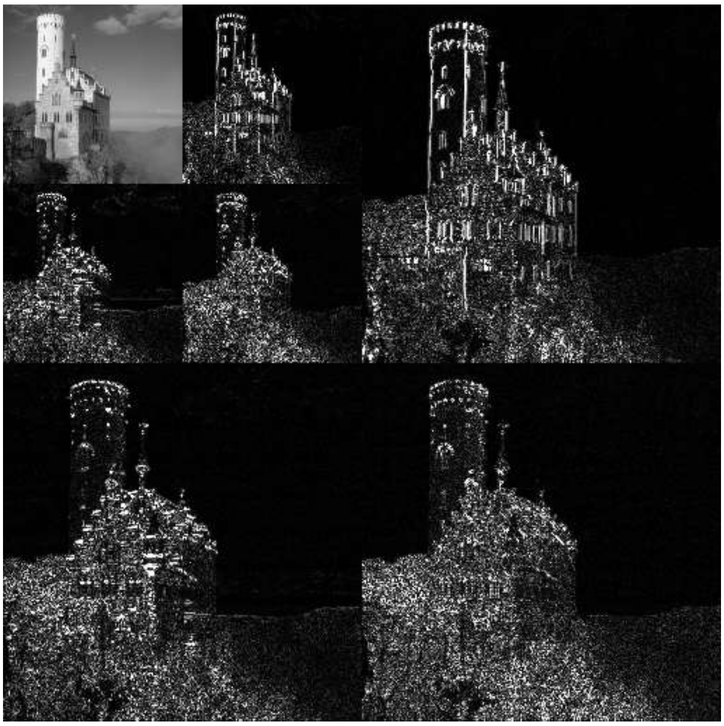


Figure 3.9 DWT Decomposition (Wikipedia, 2012).

3.3.3.5.1.3 Complex Wavelet Transforms

The discrete wavelet transform can be applied successfully to the wide range of signal and image processing tasks. However, DWT has some sort of performance limitations because of the following subjects; Oscillation, Shift variance, Aliasing and Lack of directional selectivity. Therefore, Complex wavelet transform created to perform the lack of performance of DWT. Complex wavelet transform is the modified and extended version of DWT. In addition, the complex wavelet transform employs analytic or quadrature wavelets guaranteeing magnitude-phase representation, shift invariance and no aliasing (Hostalkova, E., & Prochazva, A., 2007).

3.4 Feature Extraction and Selection

Feature Extraction is defined as locating those pixels in an image in dataset that have some distinctive attributes. Generally, that characteristic is some inhomogeneity in local image properties. (Lester, E. D., 1998)

Feature selection is the process of selecting the relevant and informative features and it can have other motivations like general data reduction, feature set reduction, performance improvement and data understanding (Guyon, I., & Elisseeff, A., 2006).

Feature extraction reduce dimensionality by (linear or nonlinear) projection of D-dimensional vector onto d-dimensional vector ($d < D$) and feature selection reduce dimensionality by selecting subset of original variables (Hulle M. V., & Davis, J., 2009). In addition, there are two methods for feature extraction and selection. These methods are unsupervised and supervised methods. Unsupervised methods are component analysis and supervised methods are classification regression (Hulle M. V., & Davis, J., 2009).

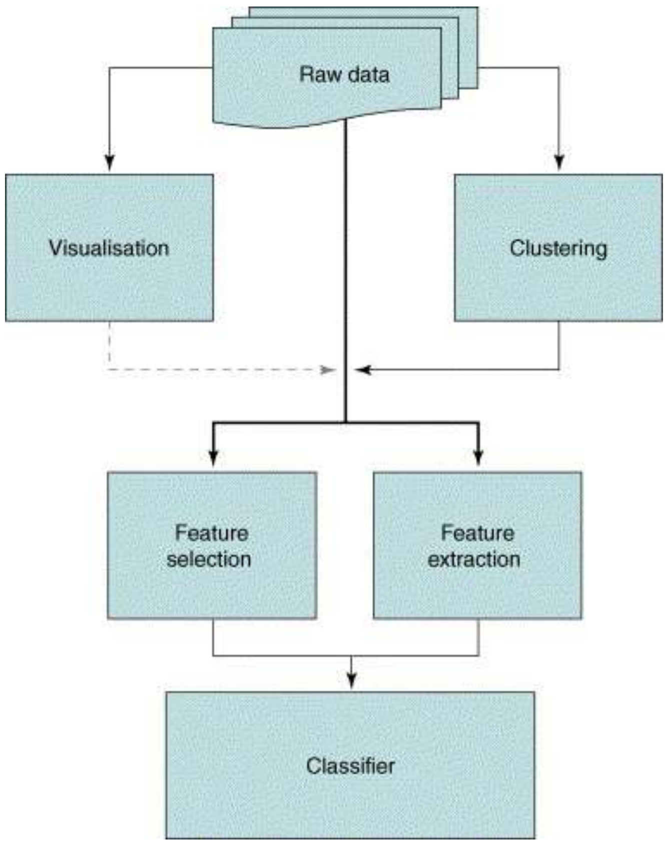


Figure 3.10 General Overview of a Classification Process with Feature Steps (Girolami, M., Mischak, H., & Krebs, R., 2006)

3.5 Classification

As mentioned above, image classification is the operation of sorting of images into a finite number of individual classes. In order to achieve this operation, there are some sorts of widely used techniques in computer artificial intelligence industry. The most successful, and advanced ones are Artificial Neural Networks (ANN), Support Vector Machines (SVM), Fuzzy Measures and Genetic Algorithms (GA) (Seetha, M., MuraliKrishna, I. V., & Deekshatulu, B. L., (n.d.)). These techniques are the leading systems in the industry because ANN can help to make better classifications by using textural features; SVM is the one of the best machine learning algorithms for classification of data sets especially in higher dimensions; Fuzzy measures locates the textures in images by using some sort of complex properties; and finally Genetic Algorithms creates space of image processing operations, generate suitable features on those spaces and classify images using that corresponding features (Seetha, M., MuraliKrishna, I. V., & Deekshatulu, B. L., (n.d.)). As a result, every technique has its own strength and weakness. In order to achieve a specific task, for looking strengths and advantages right technique could be chosen.

3.5.1 Artificial Neural Network

Artificial neural network technique is a mathematical model and improved using principle of biological neural networks of human body. Actually, it can be defined as emulation of biological neural networks. In Artificial Neural Network (ANN), simple artificial node, smallest part of the ANN, is called neuron. Those neurons are connected together to form the whole system. ANN contains group of neurons that are interconnected with each other depending on their layers and it operates information using connections to do computations. Generally, ANN has an adaptive system that changes its structure during learning stages. ANN uses model relationships between inputs and outputs or to locate patterns in data.

ANN has some advantages and disadvantages over other techniques and these advantages and disadvantages are listed below.

Advantages:

- A neural network can perform tasks that a linear program cannot.
- When an element of the neural network fails, it can continue without any problem by their parallel nature.
- A neural network learns and does not need to be reprogrammed.
- It can be implemented in any application.
- It can be implemented without any problem.

Disadvantages:

- The neural network needs training to operate.
- The architecture of a neural network is different from the architecture of microprocessors therefore needs to be emulated.
- Requires high processing time for large neural networks.

(Rios, D., 2010)

3.5.2 Support Vector Machine

Support Vector Machine (SVM) is the one of most powerful and popular classification technique of nowadays. According to the William S. Noble, "A support vector machine (SVM) is a computer algorithm that learns by example to assign labels to objects". For example, an SVM can learn to recognize handwritten digits by examining a large collection of scanned images of handwritten characters or SVMs have also been successfully applied to an increasingly wide variety of biological applications (Noble, W. S., 2006).

A SVM can create a hyperplane or set of hyperplanes in a high or infinite dimensional space, that can be used for classification, regression, or other tasks. In order to achieve good separation by the hyperplane, it has the largest distance to the nearest training data point of any class where larger margin lowers the generalization error of the classifier.

There are two types of SVMs and the mathematical formulas of these SVMs are listed below (StatSoft, Inc., 2012).

$$\frac{1}{2}w^T w + C\sum_{i=1}^{N}\xi_i \text{ subject to the constraints: } y_i(w^T\Phi(x_i) + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0, i = 1,...., N$$

$$\frac{1}{2}w^T w - v\rho + \frac{1}{N}\sum_{i=1}^{N}\xi_i \text{ subject to the constraints: } y_i(w^T\Phi(x_i) + b) \geq \rho - \xi_i \text{ and } \xi_i \geq 0, i = 1,...., N$$

There are four types of kernel functions for SVMs. Those are Linear, Polynomial, Gaussian Radial Basis Function, and Hyperbolic Tangent. In addition, mathematic formulas and figures of SVM classifications are listed below.

$$\phi = \left\{ \begin{array}{cc} x_i * x_j & Linear \\ (\gamma x_i x_j + coefficien\,t)^{\deg ree} & Polynomial \\ \exp(-\gamma \mid x_i - x_j \mid^2) & RBF \\ \tanh(\gamma x_i x_j + coefficien\,t) & Sigmoid \\ \end{array} \right\} \tag{3.4}$$

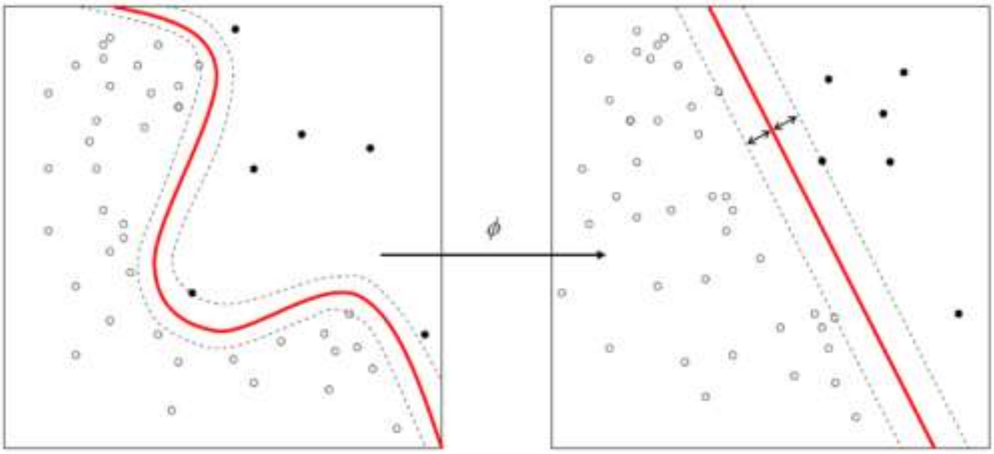


Figure 3.11 SVM Input and Feature Spaces and Classification using Kernel Functions (StatSoft, Inc., 2012).

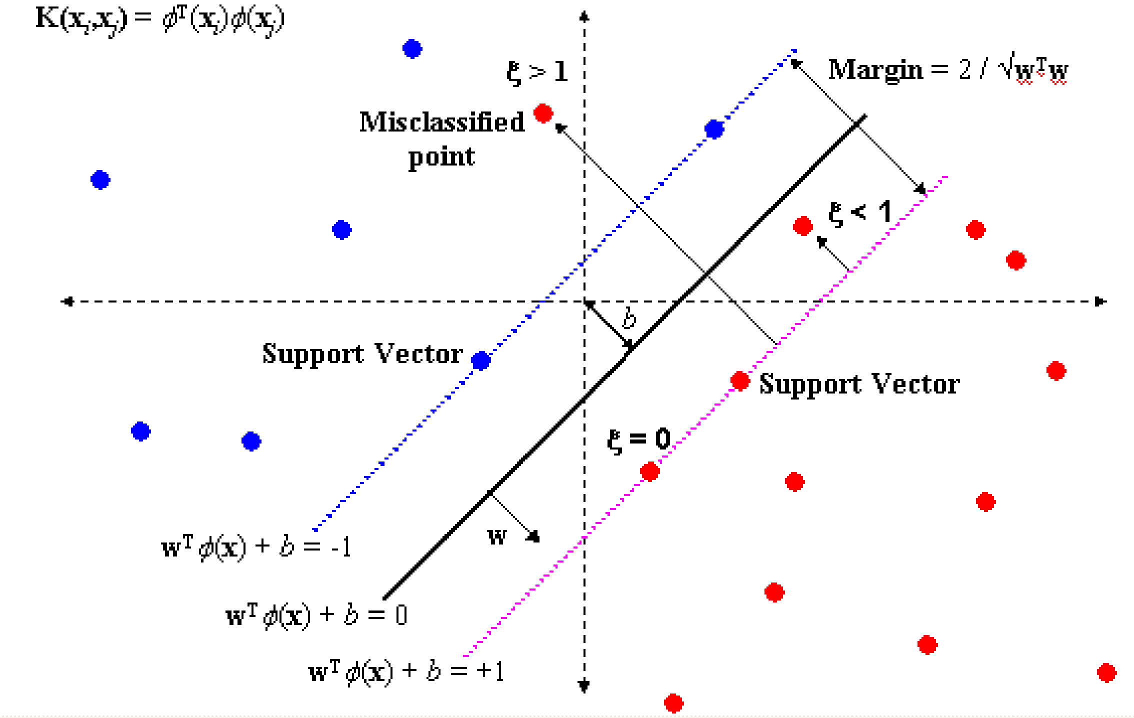


Figure 3.12 The SVM learns a hyperplane which best separates the two classes. (Microsoft Research Website, 2013).

CHAPTER 4

SUPPORT VECTOR MACHINE

4.1 Overview of SVM

The Support Vector Machine (SVM) was first introduced by Vapnik and SVM got very high interest in the field of machine learning. According to several recent studies, SVM usually has higher performance for classification issue than the other data classification algorithms. In addition, SVMs are used in wide range of real life problems like text categorization, hand-written digit recognition, tone recognition, image classification and object detection, micro-array gene expression data analysis, data classification. Moreover, SVM has some sort of strengths over the other supervised learning methods. However, performance of SVM depending on datasets can be sensitive and cost parameter and kernel function have to be set. If they are not set correctly, performance of SVM cannot be good. As a result, by using model selection process which is the process for making cross validation to find out optimal parameters, SVM can define optimal parameter properties for particular dataset. However, model selection process is very time consuming process and it can affect results of SVM easily (Srivastava, D. K., & Bhambhu, L., 2009).

According to Durgesh K. Srivastava and Lekha Bhambhu, "SVMs are set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classification. A special property of SVM is, SVM simultaneously minimize the empirical classification error and maximize the geometric margin. Therefore, SVM is called Maximum Margin Classifiers" (Srivastava, D. K., & Bhambhu, L., 2009). In addition, SVM put input vector to a higher dimensional space and construct a maximal separating hyperplane. Two parallel hyperplanes are created on each side of the hyperplane for separating the data. The separating hyperplane is the hyperplane that maximize the distance between the two parallel hyperplanes. There is an assumption that the larger margin or distance between these parallel hyperplanes makes the generalization error of the classifier better for classification results (Srivastava, D. K., & Bhambhu, L., 2009).

On the other hand, there is some important point of SVM to define it in mathematical way. First, we can consider data points in the form $\{(x_1, y_1), (x_2, y_2), (x_3, y_3)\ldots (x_n, y_n)\}$, where

$y_n=1$ / -1, a constant denoting the class to which that point x_n belongs n = number of sample and each x_n is p-dimensional real vector. The scaling is major property for protect attributes with larger variance and training data can be defined by using means of the separating hyperplane, which takes $w.x + b = o$ where b is scalar and w is p-dimensional vector. The vector w is perpendicular to the separating hyperplane and adding the offset parameter b permits the incretion of the margin. At the absent of b, the hyperplane is forced to pass via origin and it restricts the solution. In order to manage maximum margin, parallel hyperplanes of SVM have to be constructed and parallel hyperplanes can be defined by equation $w.x + b = 1$ and $w.x + b = -1$. If the training data are linearly separable, we can select these hyperplanes. That is, there are no points between them and then try to maximize their distance. By geometry, distance between the hyperplane can be found using $2 / |w|$. Therefore, to minimize $|w|$ for all excite data points, $w.\ x_i - b \geq 1$ or $w.\ x_i - b \leq -1$ should be satisfied. As a result it can be written as $y_i\ (w.\ x_i - b) \geq 1,\ 1 \leq i \leq n$ and the graph of this operation can be shown below at figure 4.1. In addition, samples along the hyperplanes are called Support Vectors (SVs) and separating hyperplane with largest margin can be defined by $M = 2 / |w|$ that is specifies support vectors means training data points closets to it and which it satisfies the equation $y_j\ [w^T.\ x_j + b] = 1\ , i = 1.$

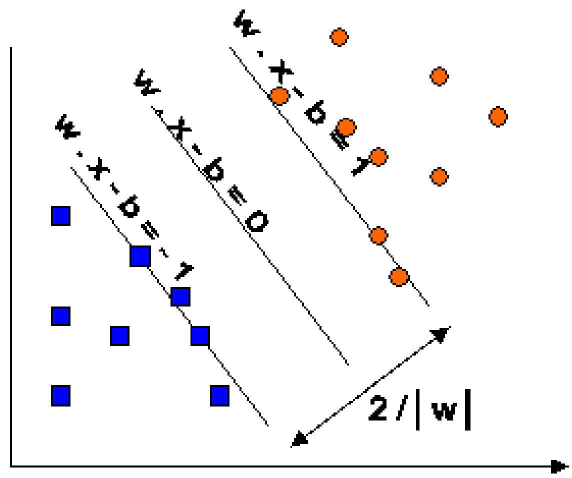


Figure 4.1 Max margin hyperplanes for a SVM with samples from two classes (Srivastava, D. K., & Bhambhu, L., 2009).

In order to find the optimal separating hyperplane having a maximum margin, a learning machine should minimize $\|w\|^2$ that inequality constraints yi $[w^T.\ x_i + b] \geq 1;\ i = 1,\ 2\ldots$ (Srivastava, D. K., & Bhambhu, L., 2009).

4.2 Kernel Methods Of SVM

At last years, Kernel methods have gained significant attention directly proportional to the increased popularity of SVM. Kernel functions can be used in different applications which they can easily transfer from linearity to non-linearity algorithms that are defined by terms of dot products. In addition, kernel methods are a group of algorithms for pattern analysis or recognition and they are the most important component of SVM. Finally, the main property of kernel methods is their unique approach for the problem. Kernel methods transform data into higher dimensional spaces to make data easily separately or create a better structure for data Souza, C. (2012).

In order to achieve classification process using SVM is depending on used kernel function. There are four types of SVM with kernel functions and those SVMs are listed below.

- Linear SVM,
- Polynomial SVM ,
  - Kernel Function: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0.$ (4.1)
- Gaussian Radial Basis Function (RBF) SVM,
  - Kernel Function: $K(x_i, x_j) = exp(-\gamma k x_i - x_j k^2), \gamma > 0.$ (4.2)
- Sigmoid SVM
  - Kernel Function: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$ (4.3)

Not: Here, γ, r, and d are kernel parameters.

(Abimanyu, N. N., Gugapriya, S., & Lalitha D., 2010)

4.2.1 Linear SVM

For a two class learning problem, the data contains objects that are labeled with one of two labels corresponding to those two classes. Thus, to make it more understandable, there is an assumption that considers the labels one is +1 positive example and other one is -1 negative example. Then, boldface x is notated as a vector with components x_i. The notation x_i will denote the i^{th} vector in a dataset, $f\ (x_i;\ y_i)\ g\ n\ i = 1$, where yi is the label associated with $x_{i.}$ In addition, the objects, $x_{i,}$ are called patterns or examples and those examples are suppose to belong the some set X.

To define a linear classifier, one thing is very important and that is dot product between two vectors. Actually, it also referred as an inner product or scalar product. So, it defined as

$w^T x = \sum_i w_i x_i$. Therefore, a linear classifier is based on a linear discriminant function of the form $f(x) = w^T x + b$. So, the vector w is weight vector and b is the bias. At first case take $b = 0$ and the set of points x such that $w^T x = 0$ are all points that are perpendicular to w and go via the origin. It can be a line in two dimensions, a plane in three dimensions, or generally, a hyperplane. The bias b translates the hyperplane away from the origin. The hyperplane $\{f(x): f(x) = w^T x + b = 0\}$ divides the space into two: the sign of the discriminant function f(x) denotes the side of the hyperplane a point is on the following figure. The boundary between regions classified as positive and negative is called the decision boundary of the classifier. The decision boundary defined by a hyperplane and it is linear because of the input examples.

As a result, linear classifier can be defined as a classifier with a linear decision boundary and conversely if the decision boundary of classifier is depending on the non-linear data, the classifier will be non-linear.

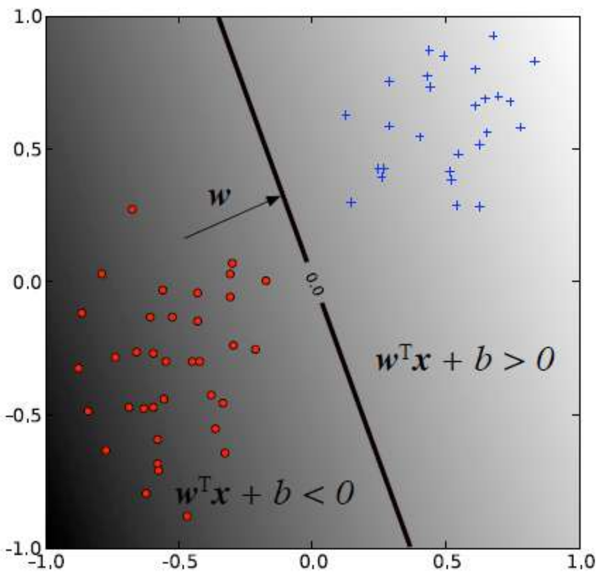


Figure 4.2. A sample of decision boundary and a linear classifier (Ben-Hur, A., & Weston, J. 2006).

4.2.2 Non-Linear SVM

First of all, non-linear classifiers generally provide better accuracy for many applications. In addition, linear classifiers have some advantages over non-linear classifiers, such as they have more simple training algorithms (Ben-Hur, A., & Weston, J. 2006).

Secondly, the correct way of making a non-linear classifier from a linear classifier is translating input space X to a feature space F using a non-linear function *ø: X -> F* (Ben-Hur, A., & Weston, J. 2006).

Finally, the discriminant function can be defined *as* $f(x) = w^T ø(x) + b$. In addition, at the upcoming sections, kernel functions will be explained to understand non-linear classifiers better and clearer (Ben-Hur, A., & Weston, J. 2006).

4.2.2.1 Polynomial Kernel Function

The polynomial kernel is the sample of non-stationary kernel and they are suitable for problems where the data in the training set is normalized (Souza, C., 2012). The formula is shown below.

$$k(x, y) = (\alpha x^T y + c)^d \tag{4.4}$$

There are some parameters that are belonged to the formula which can be configured. Those parameters are slope alpha, constant c and the degree d of the polynomial (Souza, C., 2012).

4.2.2.2 Gaussian RBF Kernel Function

The formula of Gaussian rbf kernel can be represented in two ways. Both of them listed below.

1. $$k(x, y) = \exp\left(-\frac{\| x - y \|^2}{2\sigma^2}\right) \tag{4.5}$$

2. $$k(x, y) = \exp(-\gamma \| x - y \|^2) \tag{4.6}$$

Gaussian kernel has also some parameters that can be adjustable depending on the problem. First adjustable parameter is sigma and in fact, sigma is very important for performance of the kernel. If the sigma configured in a wrong way, it affects performance of kernel in a bad way so that tuning process of sigma should be done carefully. On the other hand, if the estimation of sigma is high, exponential equation closes to linear and it starts to lose higher dimensions. However, if the estimation of sigma is low, the function loses the capability of regularization and the decision boundary becomes very sensitive to noise in training data (Ben-Hur, A., & Weston, J. 2006).

4.2.2.3 Sigmoid Kernel Function

Sigmoid kernel function also known as the Hyperbolic Tangent Kernel or Multilayer Perceptron (MLP) kernel. The motivation of sigmoid kernel comes from the field Neural Networks. The bipolar sigmoid function is used as an activation function for artificial neurons. The formula of sigmoid kernel function is illustrated below.

$$k(x, y) = \tanh(\alpha x^T y + c) \tag{4.7}$$

There are some parameters that can be changeable at the sigmoid kernel as well. First one is slope parameter alpha and the second one is constant parameter c. In addition, 1/N is a common value for alpha variable and N represents the dimension of data.

As a result, SVM model with sigmoid kernel function is the same of two layer Perceptron neutral network. Therefore, it is one of the popular kinds of SVM and works accurately in practice.

4.3 Advantages and Disadvantages of SVM

There are several advantages and few disadvantages of support vector machines over other classifiers. All strengths and weaknesses of SVM are listed below.

Strengths:

- Training is relatively easy
- No local optimal, unlike in neural networks
- It scales relatively well to high dimensional data
- Trade off between classier complexity and error can be controlled explicitly
- Non-traditional data like strings and trees can be used as input to SVM, instead of feature vectors
- By performing logistic regression (sigmoid) on the SVM output of a set of data, a SVM maps to probabilities (Maiga, A., 2011)

Weaknesses:

- Need to choose a good kernel function.
- It's time consuming (Maiga, A., 2011)

As a result, obviously SVM has more important advantages then its disadvantages. For example, SVM has two weaknesses that are requirement of selection of kernel function and it takes long time to operate. However, it is easy and it can be work on higher dimensions. To sum up, easy use, accurate performance and working on higher dimensions are more important than weakness of time consuming and kernel function selection.

4.4 SVM Training Algorithms

SVM classification process considers the standard two-class soft-margin classification that separates and classify a given data point x $\in \Re^n$ by assigning a label y $\in$ {-1, 1}. Given a defined training set consists of a dataset with data points $x_i, i \in \{1,..., l\}$ with matching labels $y_i, i \in \{1,..., l\}$. Therefore, SVM Training can be writing in the following Quadratic form, where α_i is a set of weights (one for each training point), and weights are being optimized to determine the SVM classifier. In addition, C is a parameter for managing the accuracy of training set, and $Q_{ij} = y_i y_j \Phi(x_i, x_j)$ where $\Phi(x_i, x_j)$ is a kernel function. (Campbell, C., Frieb, T. T., & Cristianini, N., (n.d.))

$$\max_{\alpha} F(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\alpha^T Q \alpha \text{ subject to } \begin{array}{l} 0 \le \alpha_i \le C, \forall_i \in 1...l \\ y^T \alpha = 0 \end{array} \tag{4.8}$$

On the other hand, to maximize a concave Lagrangian under linear constraints is gradient ascent. The formula of maximized lagrangian is shown in figure 4.9.

$$L(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \lambda \sum_i \alpha_i y_i \tag{4.9}$$

4.4.1 Kernel – Adatron Algorithm

This algorithm knows as Kernel Adatron or KA algorithm and it is based on Adatron Algorithm. However, it is adapted by the introduction of kernel function so that this algorithm can find non-linear decision boundaries using the high-dimensional feature space of SVM. By modifying the Adatron Algorithm and including kernel functions, it has an fixed point corresponding to the maximal margin hyperplane and this method is used for classification tasks with target label $y_i = \pm 1$. (Campbell, C., Frieb, T. T., & Cristianini, N., (n.d.))

Finally, the algorithm is listed below in the figure 4.3.

1. Initialize $\alpha_i = 1$ and $\theta = 0$
2. Starting from pattern *i = 1*, for labeled points (x, y) calculate:

$$z_i = \sum_{j=1}^{p} \alpha_l y_j K(x_i, x_j) - \theta \tag{4.9}$$

3. For all patterns *i* calculate $\gamma_i = y_i z_i$ and execute steps 4 to 5 below.
4. Let $\delta\alpha^i = \eta(1 - \gamma^i)$ be the proposed change to the multipliers α^i.

5.1 If $(\alpha^i + \delta\alpha^i) \leq 0$ then $\alpha^i = 0$

5.2 If $(\alpha^i + \delta\alpha^i) > 0$ then $\alpha^i \leftarrow \alpha^i + \delta\alpha^i$.

6. Calculate:

$$\theta = \frac{1}{2}\left(\min\left(z_i^{+}\right) + \max\left(z_i^{-}\right)\right)$$

Where $\left(z_i^{+}\right)$ those are patterns *i with class label* +1 *and* z_i^{-} those with class label -*1*.

7. If a maximum number of presentations of the pattern set have been exceed or the margin $m = \frac{1}{2}\left(\min\left(z_i^{+}\right) + \max\left(z_i^{-}\right)\right)$ has approached 1 then stop, otherwise return to step 2.

Figure 4.3. Kernel – Adatron Algorithm (Campbell, C., Frieb, T. T., & Cristianini, N., (n.d.))

According to Kernel – Adatron Algorithm, some points should be explained for understanding algorithm better. First one, algorithm is equivalent to performing gradient ascent in the α_i parameter space. Second one, the chosen learning rate directly affects the convergence. For example, for Gaussian kernels, the optimal value is $\eta = 1$ and convergence is in the range $0 \leq \eta \leq 2$. Third one, kernel parameter is a design choice and KA algorithm can choose the best parameters automatically by dynamically adjusting the kernel values during the learning phase. Last one, a soft margin can be implemented simply. (Campbell, C., Frieb, T. T., & Cristianini, N., (n.d.))

CHAPTER 5

DEVELOPMENT OF IMAGE CLASSIFICATION SYSTEMS USING SUPPORT VECTOR MACHINE

5.1 Overview

At this section, development process of implemented breast cancer image classification system using support vector machines are explained step by step depending on techniques used for medical image classification and those used techniques are image acquisition, image enhancement, feature extraction and classification. At the upcoming parts, from image acquisition to classification each step will be described including Matlab codes as well because this application is implemented using Matlab software.

The aim of implemented software is using image classification techniques to segment breast cancer mammographic image and detecting the tumor area of abnormal breast. In addition, at breast mammographic images tumor area has more intense area than normal parts so that suspicious region can be detected by using some sort of image classification techniques. At this software, some of important and useful image classification techniques used for image enhancement and SVM is used for image classification.

5.2 Development the flowchart of clustering algorithm

In this thesis the clustering system is designed identification of breast cancer tumors. Here the aim was to use image classification techniques to segment breast cancer mammographic image and detect the tumor area of abnormal breast. In Fig.5.1 the flowchart of clustering system is given. The inputs of the clustering system are mammogram images. These image are divided into two subsets: train set and test set. Train set is used in training stage. At first the image segmentation is performed and unwanted region is separated from the mammogram image. Then Gaussian Smoothing filtering is applied to the segmented image. This allows to decrease noise level in images. Also for enhancement of images contrast stretching, top-hat filtering and discrete wavelet transform are applied. In next stage using thresholding technique the cancer area is detected. From the cancer area the feature vector is extracted. This feature vector is input signal for classification block. After initialization of the parameters of SVM clustering model the training of the system start. Training has been done for all images taken from database. After training the SVM clustering model is obtained. In future this model is applied for clustering of test data.

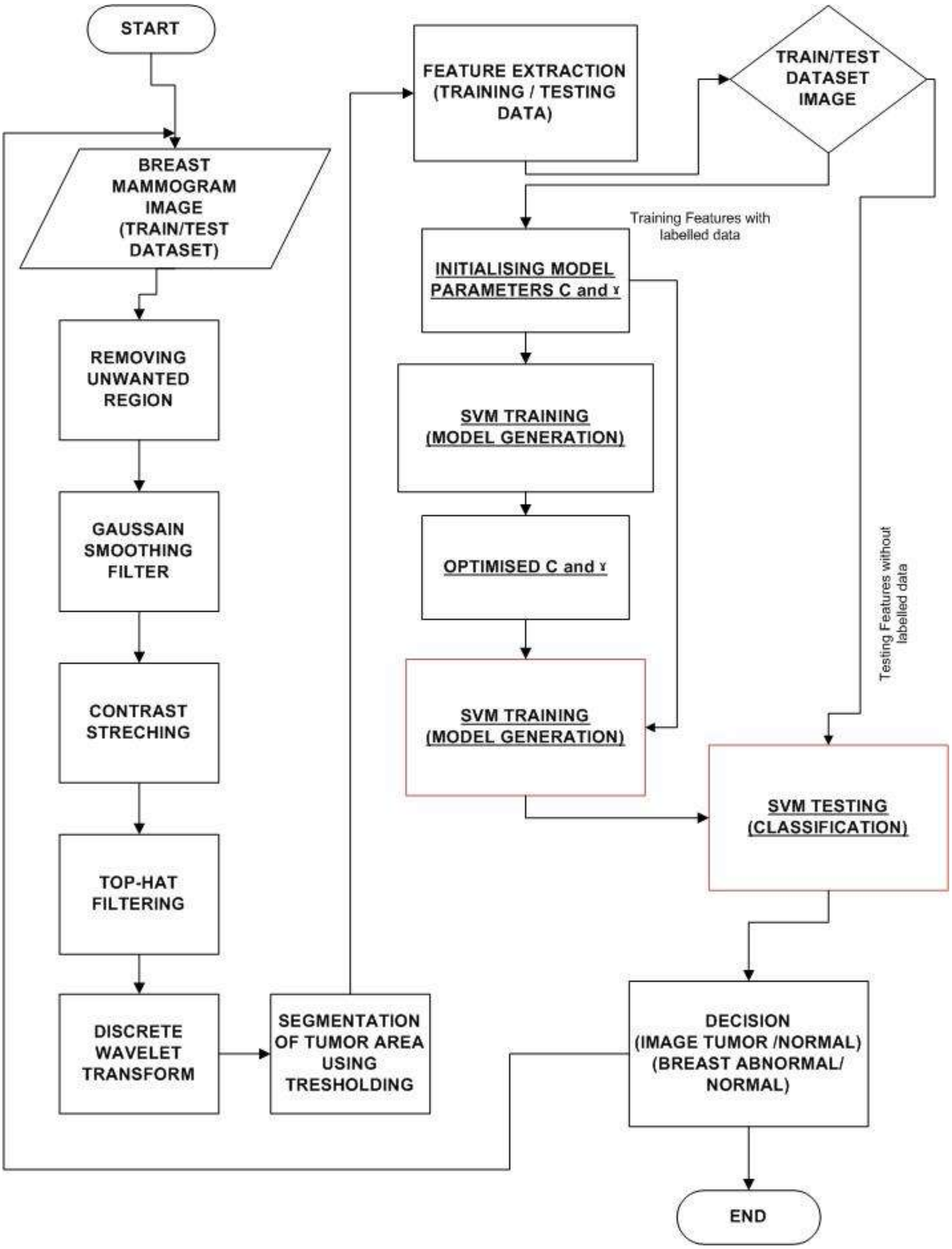


Figure 5.1 Flowchart of developed software

5.3 Image Acquisition

As mentioned above, image acquisition technique used for gathering appropriate images for classification purposes. Therefore, at this step grayscale breast images required with normal and abnormal cells that are include tumors. That is, breast cancer mammographic images very suitable. The most popular and suitable image database is "MIAS Mammographic Database" which is published by "Mammographic Image Analysis Society". As a result, at this project MIAS mammographic breast database used with normal and abnormal breast images that has breast mammographic images and they separated into two classes, normal and abnormal (tumor). In addition, MIAS mammographic image database includes 312 normal and abnormal breast images and 92 of them are used in this project. 92 images are used in this project because aim of this project is detection and classifying early stages of breast cancer so the breast images are at the early stages are chosen. In addition, there are two different set of normal data and suitable of them are chosen for image processing stages in this project.

At first section, 80 of them for training process which includes 47 abnormal and 37 normal breast mammographic images and 3 images used for test processes. At second section, 48 of them for training process which includes 27 abnormal and 21 normal breast mammographic images and 44 images used for test processes. At the following images normal and abnormal (tumor) grayscale images are shown.

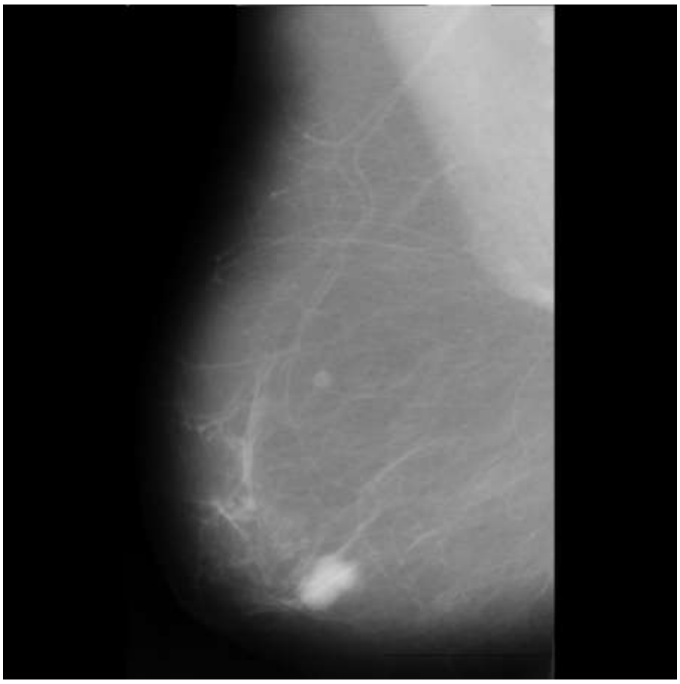


Figure 5.2 A sample of abnormal breast mammographic image

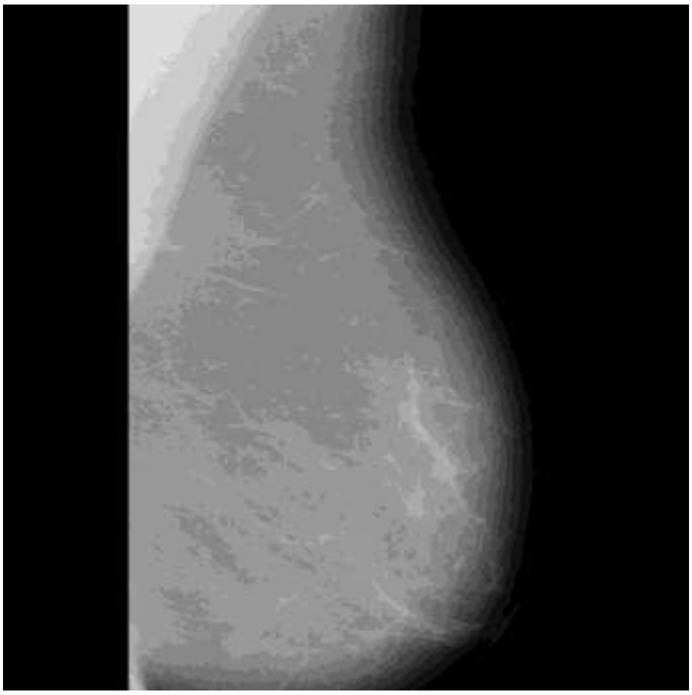


Figure 5.3 A sample of normal breast mammographic image

5.4 Image Enhancement

After choosing appropriate image formats and gathering sample images for classification purposes, image enhancement proceeded. For image enhancement, Gaussian Smoothing Filter, Contrast Stretching, Top Hat filtering, and Wavelet: Discrete Wavelet Transform.

Before using image enhancement tools to make image more clear for segmentation of tumor area and classification, an unwanted region of image is removed manually (Figure 5.4). To remove unwanted region on mammographic images, first, image converted into binary image (Figure 5.5) and big area chosen, after chosen area filled up and then using bounding box region marked and starting and finishing pixels are fixed (Figure 5.6). Finally, using those points each pixel in unwanted area converted to black and removed from the original image. Unwanted region and removed version of image (Figure 5.7) with Matlab code shown below (Figure 5.8).

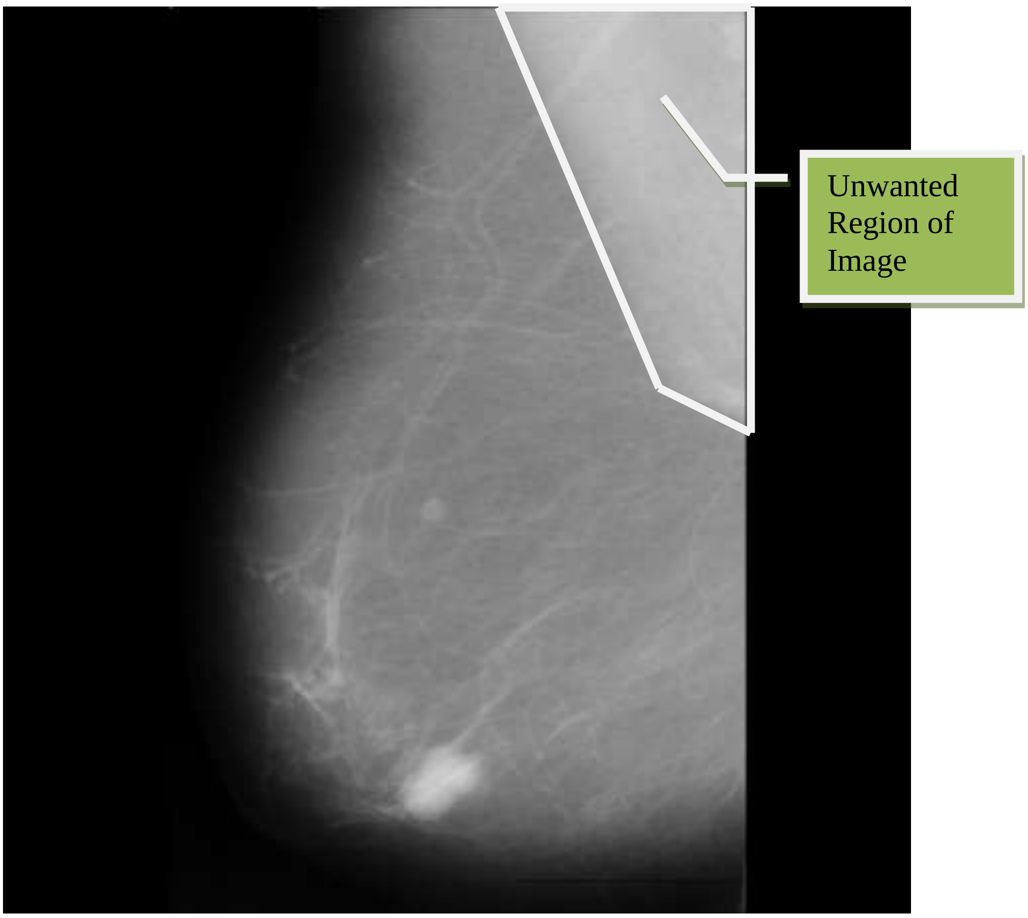


Figure 5.4 A sample of normal breast mammographic image with unwanted region

Figure 5.5 Binary image of unwanted region

Figure 5.6 Cropped image of grayscale unwanted region

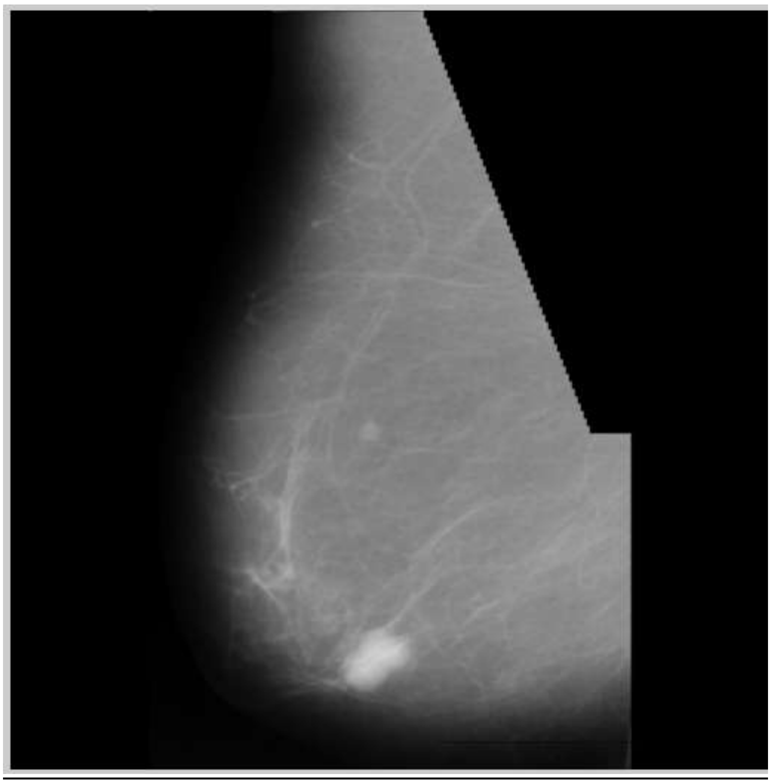


Figure 5.7 Breast Mammographic image after removing unwanted region

Matlab Code for removing unwanted region is given below;

```
ImgTTestBWU = im2bw(ImgTTest,0.70);
figure(2);imshow(ImgTTestBWU,'InitialMagnification',50);
title('Binary Version of Original Image');
pause(0.5); close();
ImgBWUSeg = bwareaopen(ImgTTestBWU, 10000);
figure(3);imshow(ImgBWUSeg,'InitialMagnification',50);
title('Binary Version Areas of Original Image');
pause(0.5); close();
ImgBWSegUFil= imfill(ImgBWUSeg,'holes');
figure(4);imshow(ImgBWSegUFil,'InitialMagnification',50);
title('Filled up Areas of B/W Image');
pause(0.5); close();
ImgStats = regionprops(ImgBWSegUFil, 'BoundingBox');
ImgStatSize=size(ImgStats);
ImgStatSizeX=ImgStatSize(1);
for ii=1:ImgStatSizeX
  BoxList{ii}=ImgStats(ii).BoundingBox;
  if((ImgStats(ii).BoundingBox(2))<10)
         selectedBoxIdx=ii;
         selectedBox=BoxList{ii};
      end
end
ImgUCrop= imcrop(ImgTTest,ImgStats(selectedBoxIdx).BoundingBox);
figure(5);imshow(ImgUCrop,'InitialMagnification',50);
title('Cropped Region of Original Image');
pause(0.5); close();
x=real(zeros); y=real(zeros); xl=real(zeros); yl=real(zeros);
xp=real(zeros); yp=real(zeros); x=(round(ImgStats(ii).BoundingBox(1)))-30;
y=(round(ImgStats(ii).BoundingBox(2)));
xl=x+(round(ImgStats(ii).BoundingBox(3)))+30;
yl=y+(round(ImgStats(ii).BoundingBox(4)))+150;
xp=x+1; yp=y+10; xloop=xl-x; yloop=(yl-y)/(yp-y);

if((round(ImgStats(ii).BoundingBox(1)))>300)
  for j=1:yloop
        ImgTTest(y:yp,xp:xl)=0;
        xp=xp+4;
        y=y+10;       yp=yp+10;
    end
else
  for j=1:yloop
     ImgTTest(y:yp,xp:xl)=0;
     xl=xl-4;
     y=y+10;    yp=yp+10;
  end
end
```

Figure 5.8 Matlab code of removing unwanted region

Secondly, Gaussian Smoothing Filter is used to remove details and noise on original image (Figure 5.9). After applying Gaussian smoothing filter, new images created and noise and details are removed. In addition, Gaussian filter is used to increase the signal to noise ratio Sample of those images and Matlab codes are listed in Figure 5.10.

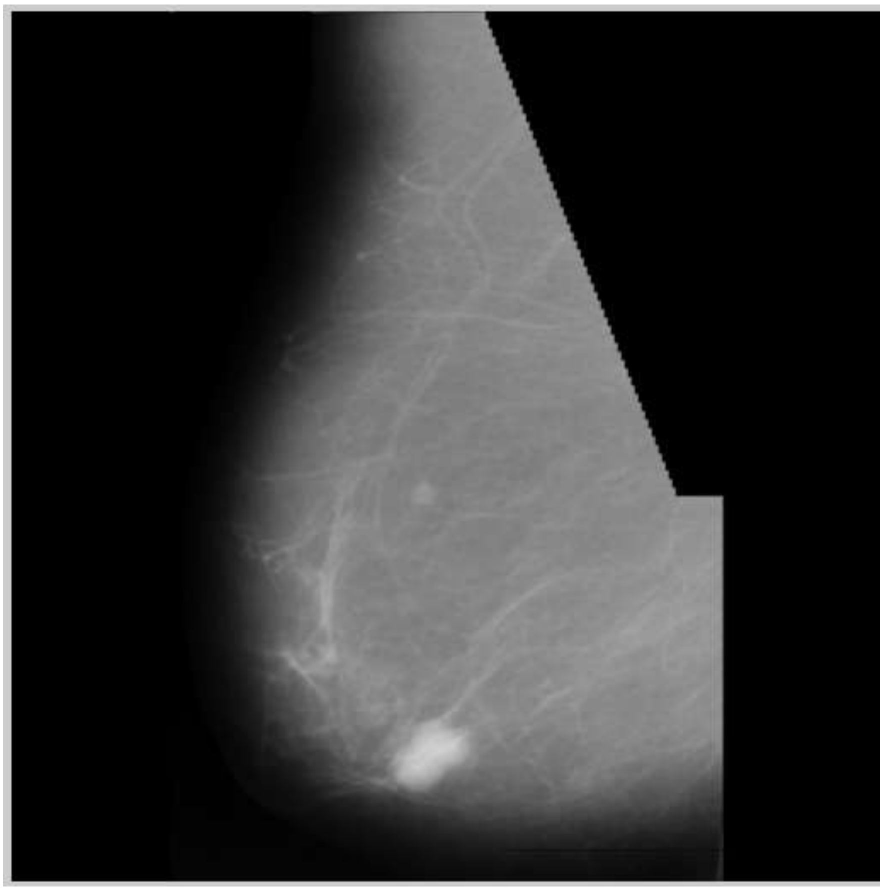


Figure 5.9 Gaussian Smoothing Filtered Image of abnormal breast after unwanted region removed from original image

Matlab Code for gaussian smoothing filter is given below

```
h = fspecial('gaussian', hsize, sigma);
blurred = imfilter(I,H,'replicate');
HTTr = fspecial('gaussian',[5 5],1);
ImgGFTT = imfilter(ImgTT,HTTr,'replicate','same');
```

Figure 5.10 Matlab code of gaussian smoothing filter

In third step, Contrast Stretching is used to improve image by stretching the range of intensity values to make colors of image see easily (Figure 5.11). Sample of those images and Matlab codes are listed in Figure 5.12.

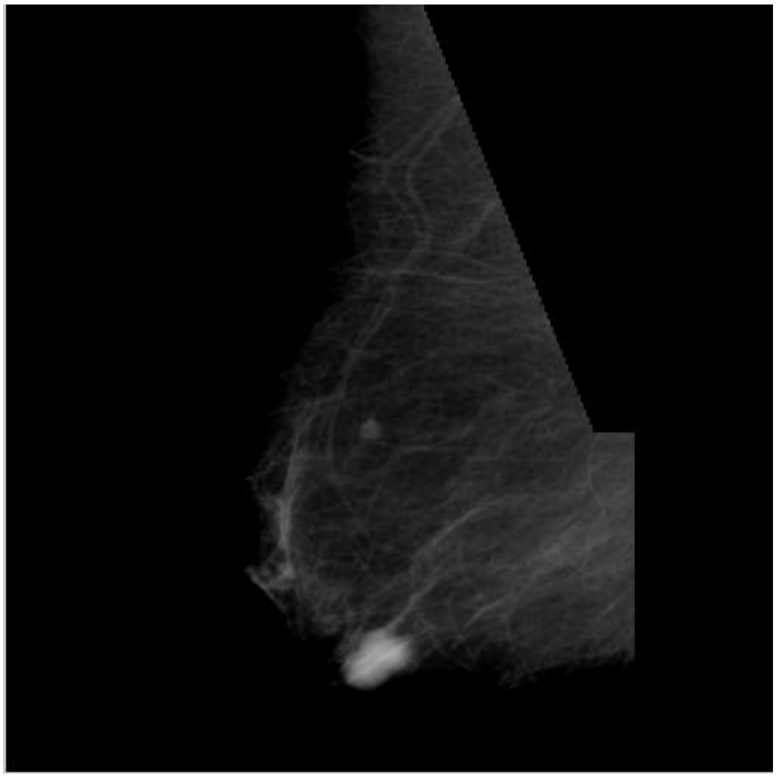


Figure 5.11 Contrast Stretched Image of abnormal breast

```
%Matlab Code for contrast stretching is given below
J = imadjust(I,stretchlim(I),[]);
ImgCSTTest = imadjust(ImgGFTTest,[0.45 1],[]);
```

Figure 5.12 Matlab code for contrast stretching

Fourthly, Top Hat Filtering is used to remove the uneven background from the image (Figure 5.13). Sample of those images and Matlab codes are listed in Figure 5.14.

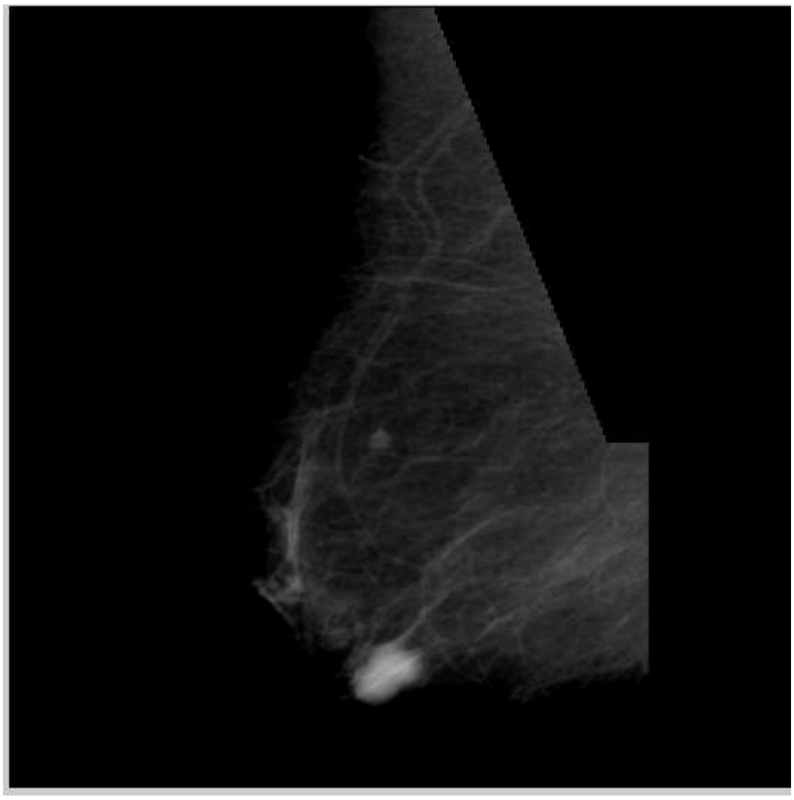


Figure 5.13 Top Hat Filtered Image of abnormal breast

```
%Matlab Code for top-hat filter is given below
J = imtophat(I,se);
seTTest = strel('disk',250);
ImgTHFTTest = imtophat(ImgCSTTest,seTTest);
```

Figure 5.14 Matlab code for top-hat filter

Fifthly, Discrete Wavelet Transform is used to decompose the image and then reconstructed it. The main aim of this process is to compress input image and reduce the time of processes.

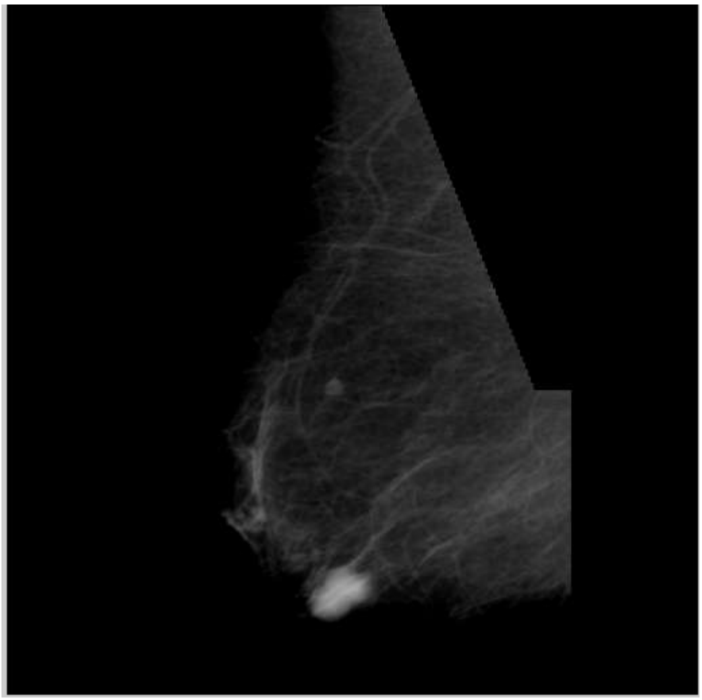


Figure 5.15 Discrete Wavelet Transform Image of abnormal breast

Matlab Code for discrete wavelet transform is given below

```
Xtest=ImgTHFTT;
N=2; [C,S] = wavedec2(Xtest,N,'db1');
cA1 = appcoef2(C,S,'db1',1); [cH1,cV1,cD1] = detcoef2('all',C,S,1);
cA2 = appcoef2(C,S,'db1',2); [cH2,cV2,cD2] = detcoef2('all',C,S,2);
A1 = wrcoef2('a',C,S,'db1',1); A2 = wrcoef2('a',C,S,'db1',2);
H1 = wrcoef2('h',C,S,'db1',1); H2 = wrcoef2('h',C,S,'db1',2);
V1 = wrcoef2('v',C,S,'db1',1); V2 = wrcoef2('v',C,S,'db1',2);
D1 = wrcoef2('d',C,S,'db1',1); D2 = wrcoef2('d',C,S,'db1',2);
 [thr,sorh,keepapp] = ddencmp('cmp','wv',Xtest);
[XC,CXC,LXC,PERF0,PERFL2]=wdencmp('gbl',C,S,'db1',N,thr,sorh,keepapp);
M=waverec2(CXC,LXC,'db1'); out1tr=uint8(M);
input_ima1=double(Xtest);  out2=double(out1tr);     error=0;
     for y=1:191
      for x=1:159
        MSE=((input_ima1(x,y))-(out2(x,y)))^2;
          error=MSE+error;
      end
     end
MSE_WO=(1/(159*191))*error;
PSNR_WO=20*log10(255/sqrt(MSE_WO));
diff_ima=imsubtract(input_ima1,out2);
figure(9);  imshow(out1tr,'InitialMagnification',50);
title('Image After Discrete Wavelet Transform');
pause(0.5); close();
```

Figure 5.16 Matlab code for discrete wavelet transform

Sixth, regional properties tool of Matlab "Regionprop" used to define properties of segmented area and selection process of each segmented area (Figure 5.17). An example of segmented tumor area is shown below and Matlab code listed as well.

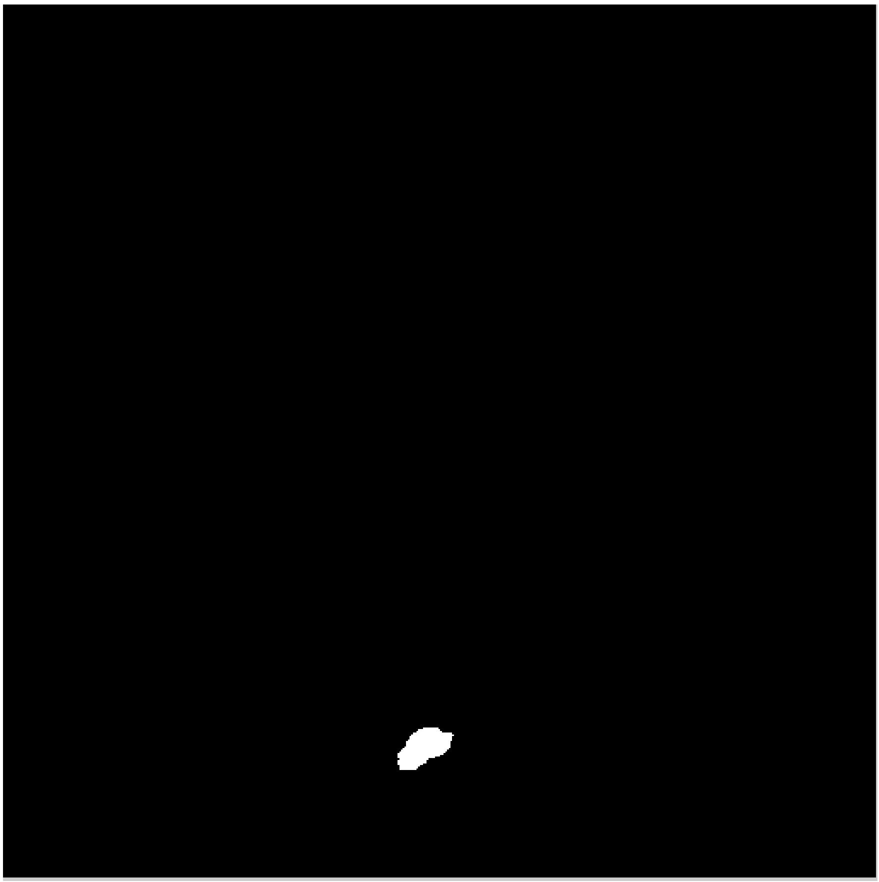


Figure 5.17 Segmented Image after Image Classification Techniques applied

Matlab Code for region segmentation is given below

```
BWLabelTTr = bwlabel(ImgGFBWTTr);
stats = regionprops(BWLabelTTr,'all');
range=300:7500;
selectedvalue=max(arealist(arealist>=min(range) & arealist<=max(range)));
if isempty(selectedvalue)
   range=7501:800000;
   selectedvalue=max(arealist(arealist>=min(range) & arealist<=max(range)));  end
```

Figure 5.18 Matlab code for region segmentation

Finally, at this section all the subheadings of the image enhancement steps are listed and explained detailed. In addition, all the images and Matlab codes are shown as well.

5.5 Feature Extraction and Selection

After completing image enhancement process of images, features are extracted and appropriate ones are selected from edited image’s tumor areas.

At Matlab, properties of segmented areas can be obtained by using regionprop function. In addition, properties of segmented tumor area are categories into two groups that are shape measurements and pixel value measurements. Therefore, main properties of tumor area are listed below.

Shape Measurements;

- Area
- EulerNumber
- Orientation
- Bounding Box
- Extent
- Perimeter
- Centroid
- Extrema
- PixelIdxList
- ConvexArea
- FilledArea
- PixelList
- ConvexHull
- FilledImage
- Solidity
- ConvexImage
- SubarrayIdx
- Eccentricity
- MajorAxisLength
- MinorAxisLength
- EquivDiameter

Pixel Value Measurements;

- MaxIntensity
- MinIntensity
- WeightedCentroid
- MeanIntensity
- PixelValues

At the above listed features of segmented area, features that are listed below are selected.

- Area: It is a scalar value and it represents the actual number of pixels in the selected region.

- Centroid: It specifies the center of mass of the region. In addition, returned value of Centroid contains horizontal coordinate (x-coordinate) and vertical coordinates.

- Major axis length: It is a scalar value and it is the length, in pixel, of the major axis of the ellipse which has same normalized central moments as the region.

- Minor axis length: It is a scalar value and it is the length, in pixel, of the minor axis of the ellipse which has same normalized central moments as the region.

- Eccentricity: It is a scalar value and it has the same second-moments as the region. The eccentricity is the ratio of the distance between the foci of the ellipse and its major axis length. In addition, the value of eccentricity is between 0 and 1.

- Orientation: It is a scalar value and it is the angle between the x-axis and the major axis of the ellipse that has the same second-moments as the region.

- Filled area: It is a scalar value and it specifies the amount of on pixels in FilledImage.

• Solidity: It is a scalar value and it specifies the proportion of the pixels in the convex hull and that are also in the region. In addition, solidity is computed as using Area/ConvexArea.

• Equivdiameter: It is a scalar value and it specifies the number of objects in the region minus the number of holes in those objects.

5.6 Classification Step:

Classification is the latest step of my application and it is used to classify image that have tumor and not. At this step, SVM used to make classification process. SVM is a supervised image classification algorithm. Because of supervised classification, dataset is apart into two groups that are labeled as training and test. In addition, to make classification process more accurate and effective, SVM used as non-linear because image dataset properties are very complex and it needed non-linear classification.

Matlab Code for SVM training and test processes are given in Figure 5.18.

5.6.1 Usage of Support Vector Machines

At this project, support vector machine is used as a classifier. There are three stages of SVM classification process. Those processes are train, test and classification. At the train process, predefined training dataset images are proceed and they were marked at the hyper plane of SVM classifier. Training process is done by using following Matlab code in Figure 5.19.

```
"svmStruct1 = svmtrain(traindata,group,'Kernel_Function','quadratic','method','QP','showplot',true); "
```

SVM training process matlab code includes two main parameters to define details like method and performance of the training process. First one is the kernel function which is used for mapping the training data to the kernel space and second one is the method for finding the separating hyperplane between classes.

Kernel Functions are described at the previous section so that at this section only the quadratic kernel function will be explained which is the kernel function of this project training process. Quadratic kernel function is a second degree polynomial kernel function so the formula for quadratic function is shown at the formula 5.1.

$$k(x, y) = (\alpha x^T y + c)^d \quad \text{α:slope, c:constant, and d: degree of the polynomial} \tag{5.1}$$

As a result, quadratic kernel function is a second degree polynomial kernel function and the formula at the above, works for mapping training data to the feature space and quadratic programming (QP) method construct the separating hyper plane for feature space. Quadratic Kernel Function and QP method is chosen for parameters of the SVM training because they are the most suitable for dataset that is used at this project.

Quadratic programming method is the one of the methods that are used in Matlab for separating hyper plane. Briefly, the classifier is a 2-norm soft-margin SVM and gives the quadratic programming options.

Finally, the formula that works at train process of Matlab is shown in equation (5.2).

$$c = \sum_i \alpha_i k(s_i, x) + b \tag{5.2}$$

Matlab train process uses above formula for train process and it uses optimization method to define support vectors si, weights αi, and bias b that are used to classify vectors x according to the above equation (5.2).

In addition, at equation (5.2) k is the kernel function and in the case of a linear kernel, k is the dot product and , if c ≥ 0, then x is classified as a member of the first group, otherwise it is classified as a member of the second group.

On the other hand, test and classification processes are done by using following Matlab code in Figure 5.19.

```
"BrstSampleList = svmclassify(svmStruct1,testData,'showplot',true) plot(testData(1),testData(2),'ro','MarkerSize',12)"
```

Main difference of test and classification processes is test process needed images more than one and classification processes only have one image because it checks the classification of only one new image. In addition, this process has no more parameters because it uses the same ones as training process.

5.6.2 Results of Image Classification with Train and Test Processes

At this section, train and test processes of developed software are done in two sections. At first section, there are 80 images for train process. Those images are includes both tumor and normal image classes. Tumor class has images that include tumor area and normal class has no images that include tumor areas. In addition, test process has three images one from tumor class and two others are from normal class because normal image class includes two different image types that are normal, no tumor area in images. At the second section, 92 images used and 48 of them for training processes and 44 of them for test processes.

As a result, images are shown at the upcoming parts.

```
BrstSample = java_array('java.lang.String', 80);
% number of photos in the folder

BrstSample(1) = java.lang.String('TUMOR');
BrstSample(2) = java.lang.String('TUMOR');
…
…
BrstSample(40) = java.lang.String('TUMOR');
BrstSample(41) = java.lang.String('TUMOR');
…
BrstSample(60) = java.lang.String('NORMAL');
BrstSample(61) = java.lang.String('NORMAL');
…
BrstSample(79) = java.lang.String('NORMAL');
BrstSample(80) = java.lang.String('NORMAL');

BrstSampleList = cell(BrstSample);

FeatureListTT =
horzcat(SImgAreaTT,SImgFilledAreaTT,SImgCentroidXTT,SImgCentroidYTT,SImgMajorAxisLengthTT,SImgMinorAxisLengthTT,SImgEccentricityTT,SImgOrientationTT,SImgSolidityTT,SImgEquivDiameterTT);

traindata=FeatureListTT(1:end,1:2);%% According to feature 1 and 2.
group=BrstSampleList(1:end);

%%%Training process SVM code
figure(8);
svmStruct1 = svmtrain(traindata,group,'Kernel_Function','quadratic','method','QP','showplot',true);

FeatureListTTest =
horzcat(SImgAreaTTest,SImgFilledAreaTTest,SImgCentroidXTTest,SImgCentroidYTTest,SImgMajorAxisLengthTTest,SImgMinorAxisLengthTTest,SImgEccentricityTTest,SImgOrientationTTest,SImgSolidityTTest,SImgEquivDiameterTTest);

testData=[FeatureListTTest(1,8),FeatureListTTest(1,9)];

%testData=[FeatureListTTest(1,1),FeatureListTTest(1,2),FeatureListTTest(1,3),FeatureListTTest(1,4),FeatureListTTest(1,5),FeatureListTTest(1,6),FeatureListTTest(1,7),FeatureListTTest(1,8),FeatureListTTest(1,9),FeatureListTTest(1,10)];
%testData=FeatureListTTest;
```

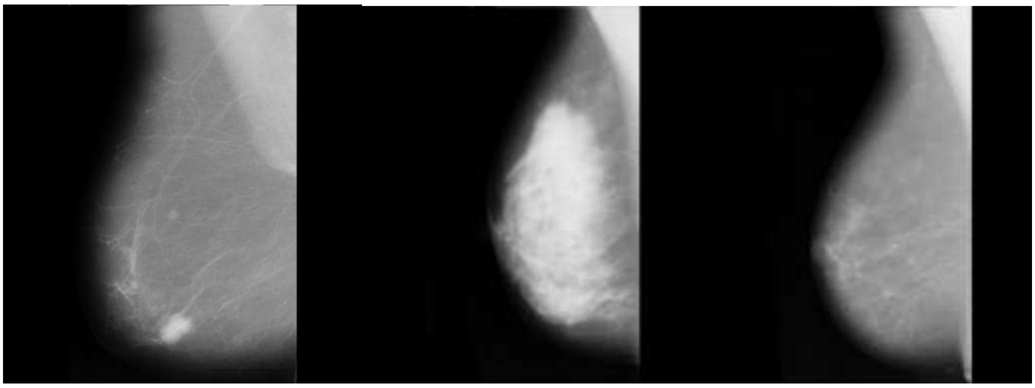
```
%%%Test/Classify process SVM code
        BrstSampleList = svmclassify(svmStruct1,testData,'showplot',true)
        hold on;plot(testData(1),testData(2),'ro','MarkerSize',12);hold off

        %hold
on;plot(testData(1),testData(2),testData(3),testData(4),testData(5),testData(6),testData(7),testData(8),testData(9)
,testData(10),'ro','MarkerSize',12);
        hold off
```

Figure 5.19 Matlab Code for SVM training and test processes

5.6.2.1 Section 1: Results of Image Classification with Train and Test Processes

Input images are taken for classification. For these images features are selected. The images are given in Figure 5.20. Using different features the training and classification of images have been performed. At first SVM training algorithm is called for input images, then classification has been performed. The Input Images and features are:



(a) (b) (c)

Figure 5.20 Image classification using SVM Test Process Input Images

(a) Abnormal Breast: It has tumor.

(b) Normal Breast: It has no tumor.

(c) Normal Breast: It has no tumor.

Features:

i-	Area	(Feature 1)
ii-	Filled Area	(Feature 2)
iii-	Centroid X	(Feature 3)
iv-	Centroid Y	(Feature 4)
v-	Major Axis Length	(Feature 5)
vi-	Minor Axis Length	(Feature 6)
vii-	Eccentricity	(Feature 7)
viii-	Orientation	(Feature 8)
ix-	Solidity	(Feature 9)
x-	EquivDiameter	(Feature 10)

5.6.2.1.1 Feature 1 and 2's Train and Test Processes with 3 Input Images

Results of training and classification for image are shown in Figure 5.21-5.24.

Classification Result:

- (a) image: image has a "**TUMOR**" and it is an abnormal breast.
- (b) image: image has no tumor and it is a "**NORMAL**" breast.
- (c) image: image has no tumor and it is a "**NORMAL**" breast.

Train Process:

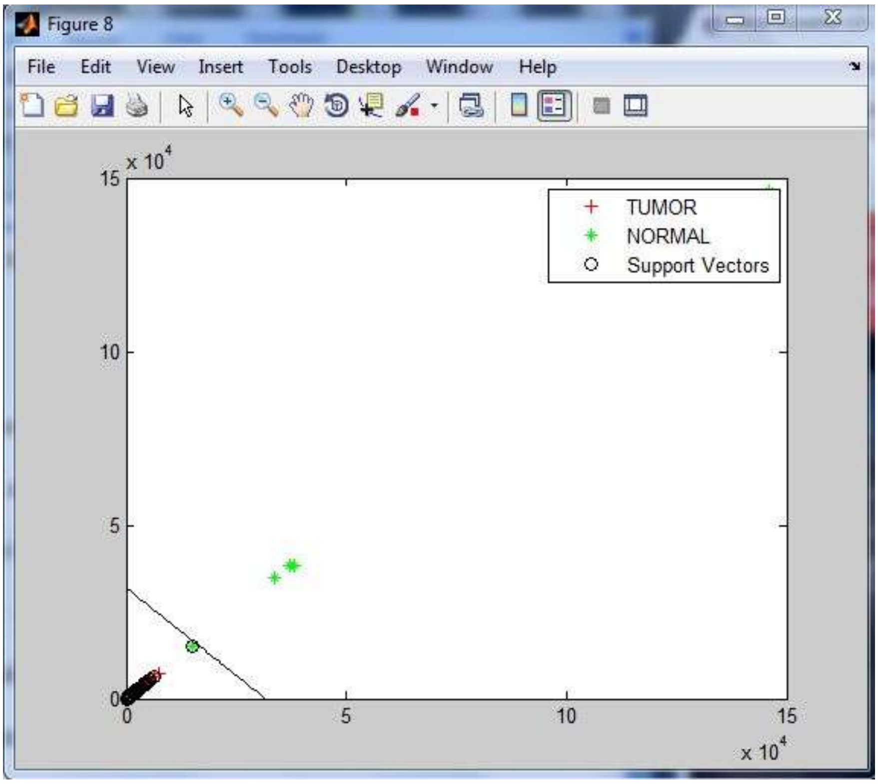
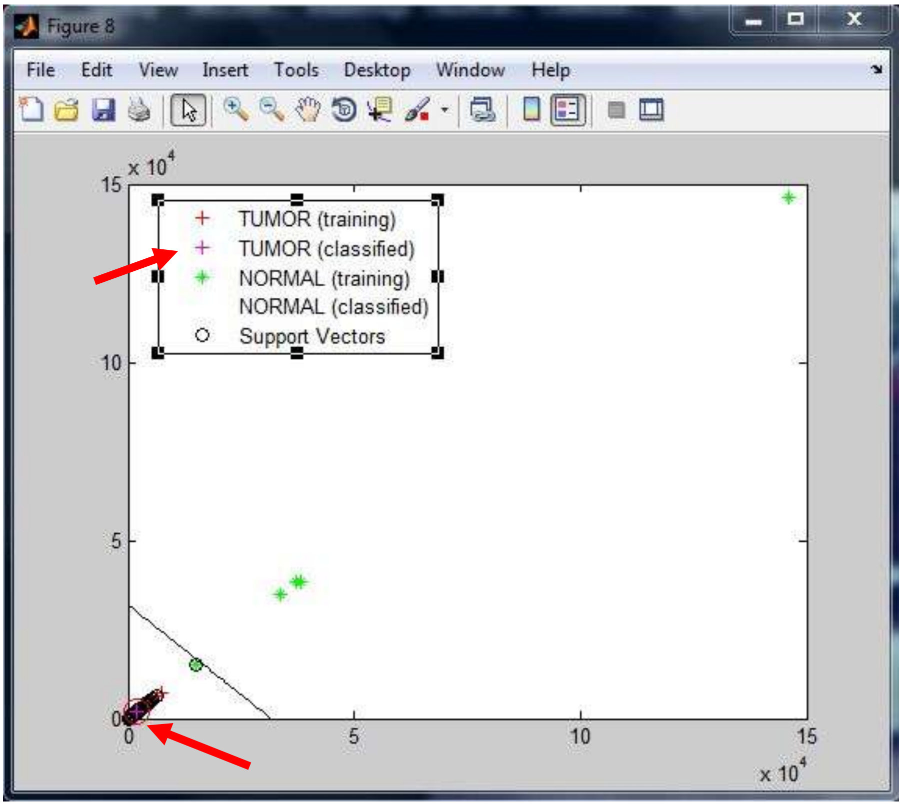


Figure 5.21 Image classification using SVM Train Process, Feature 1 2

Test Process with image A:



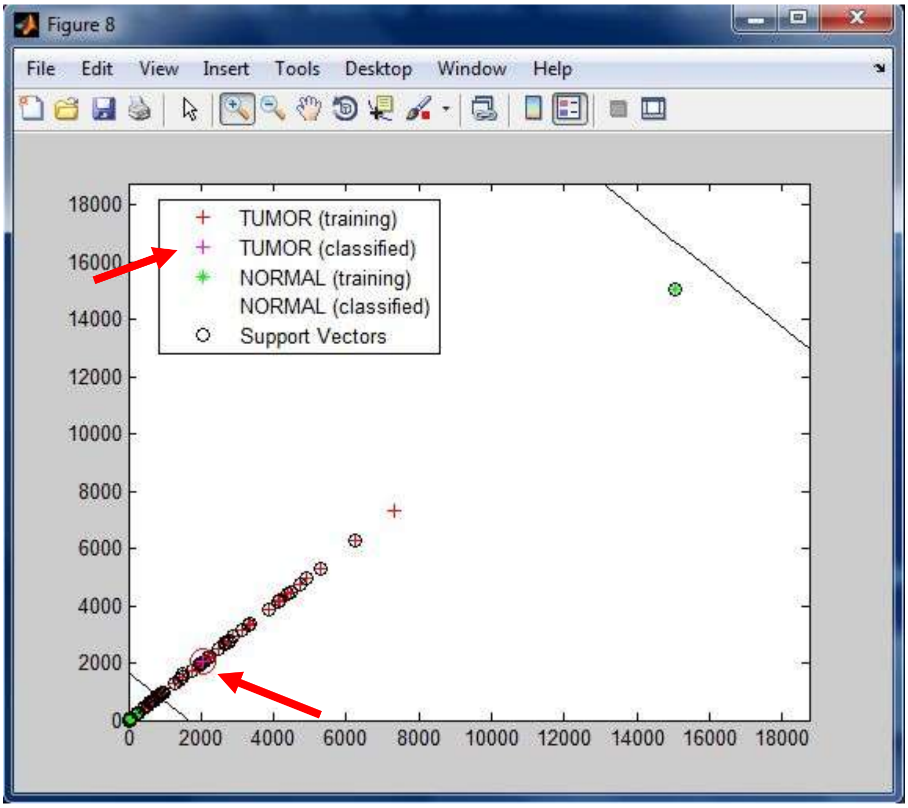


Figure 5.22 Image classification using SVM Test Process, Feature 1 2, Test Image (a)

Test Process with image B:

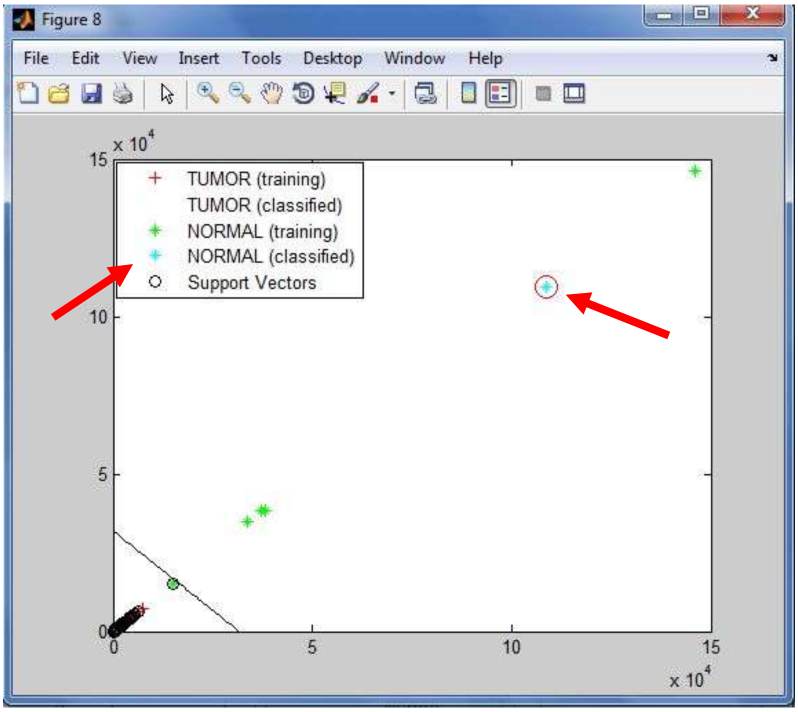


Figure 5.23 Image classification using SVM Test Process, Feature 1 2, Test Image (b)

Test Process with image C:

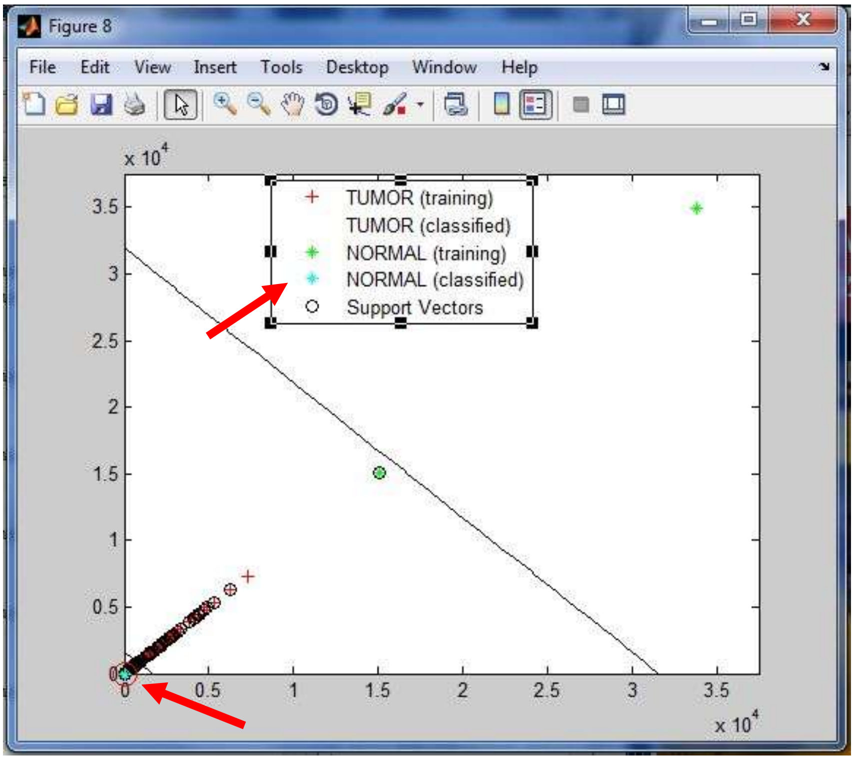


Figure 5.24 Image classification using SVM Test Process, Feature 1 2, Test Image (c)

5.6.2.1.2 Feature 3 and 4's Train and Test Processes with 3 Input Images

Results of training and classification for image are shown in Figure 5.25-5.28.

Classification Result:

- (a) image: image has a "**TUMOR**" and it is an abnormal breast.
- (b) image: image has no tumor and it is a "**NORMAL**" breast.
- (c) image: image has no tumor and it is a "**NORMAL**" breast.

Train Process:

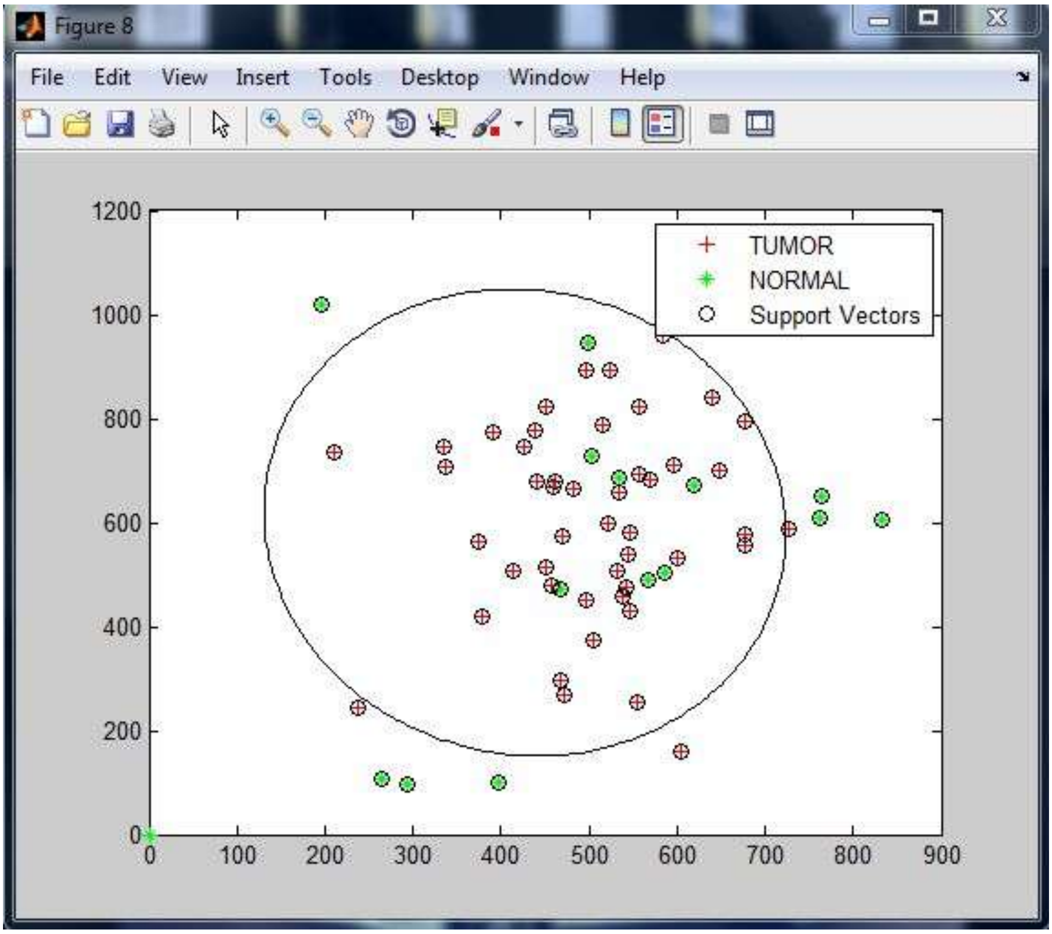


Figure 5.25 Image classification using SVM Train Process, Feature 3 4

Test Process with image A:

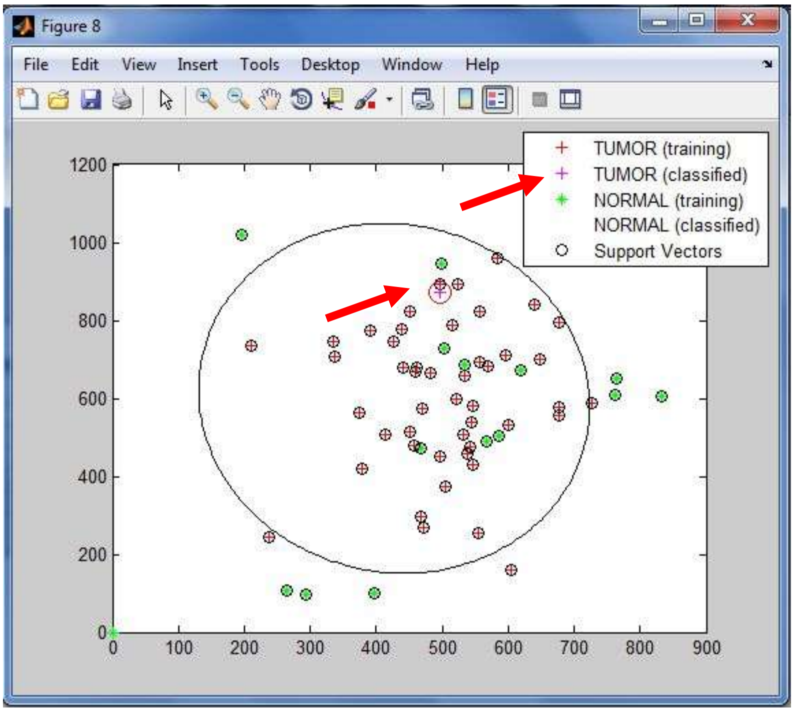


Figure 5.26 Image classification using SVM Test Process, Feature 3 4, Test Image (a)

Test Process with image B:

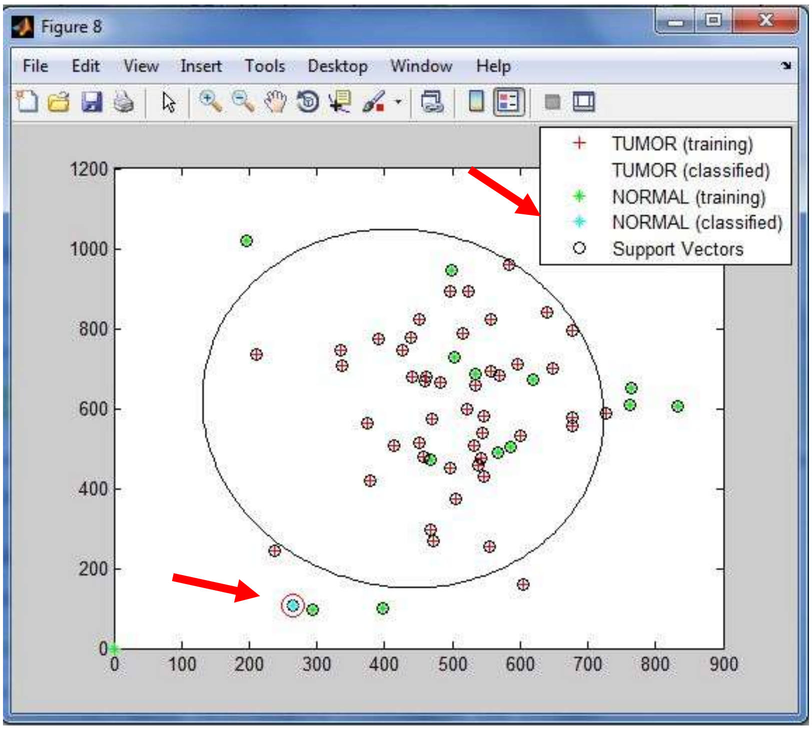


Figure 5.27 Image classification using SVM Test Process, Feature 3 4, Test Image (b)

Test Process with image C:

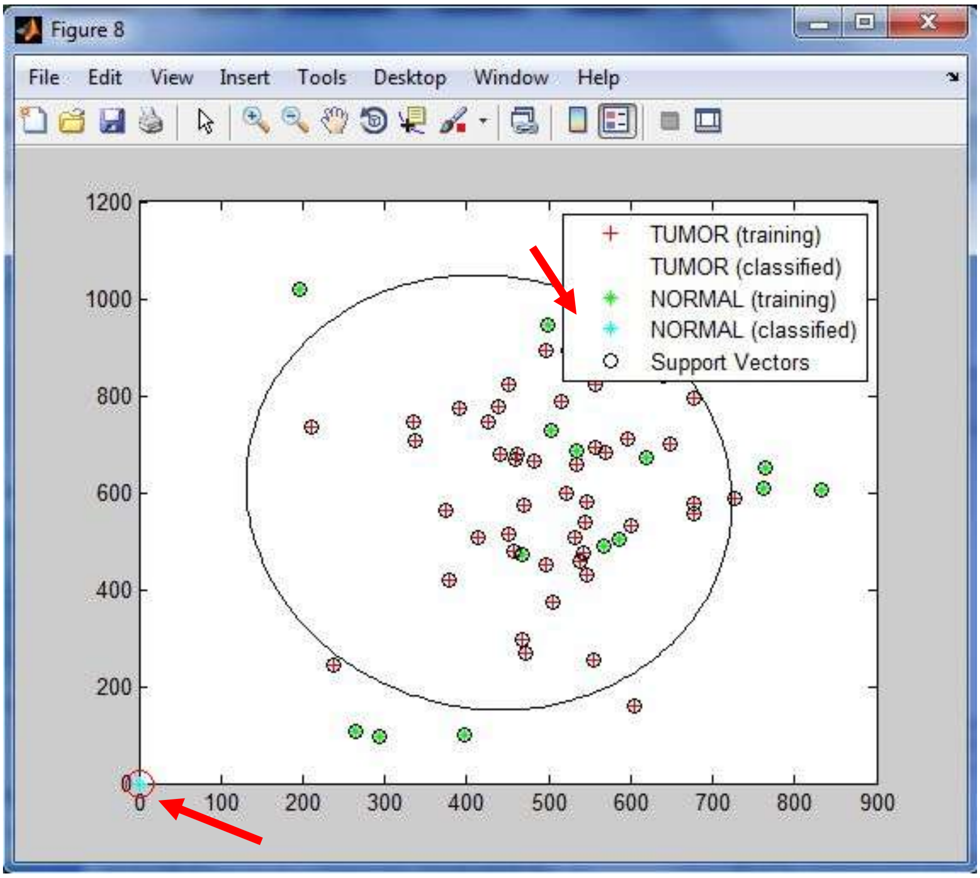


Figure 5.28 Image classification using SVM Test Process, Feature 3 4, Test Image (c)

5.6.2.1.3 Feature 5 and 6's Train and Test Processes with 3 Input Images

Results of training and classification for image are shown in Figure 5.29-5.32.

Classification Result:

- (a) image: image has a "**TUMOR**" and it is an abnormal breast.
- (b) image: image has no tumor and it is a "**NORMAL**" breast.
- (c) image: image has no tumor and it is a "**NORMAL**" breast.

Train Process:

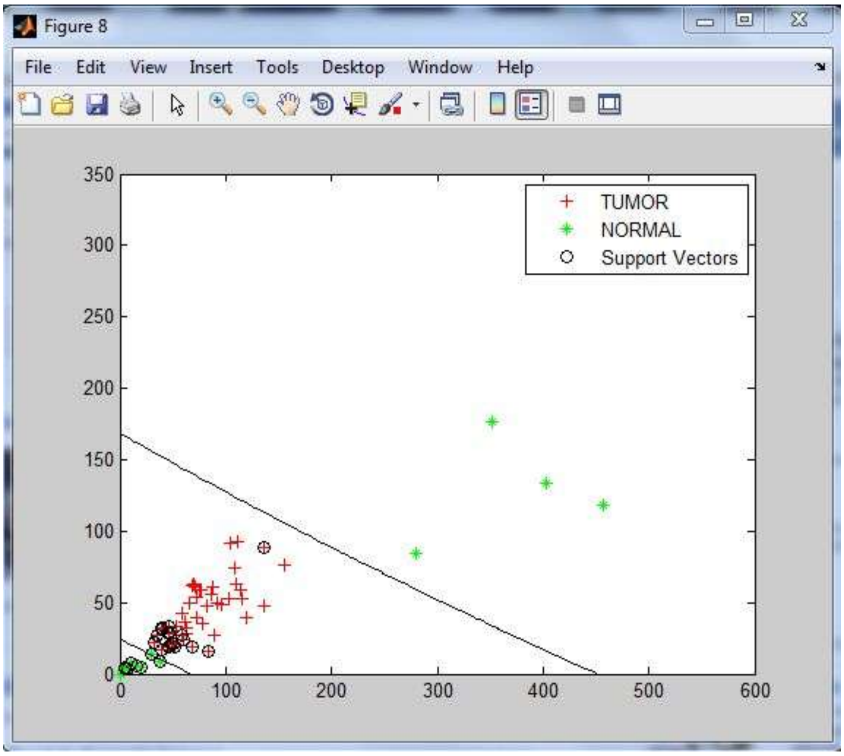


Figure 5.29 Image classification using SVM Train Process, Feature 5 6

Test Process with image A:

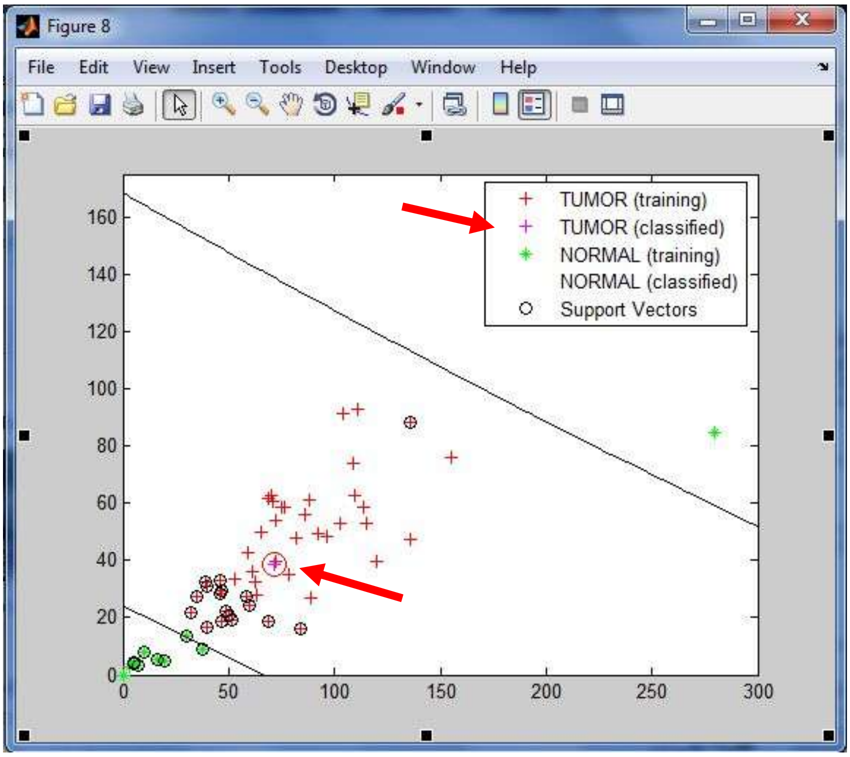


Figure 5.30 Image classification using SVM Test Process, Feature 5 6, Test Image (a)

Test Process with image B:

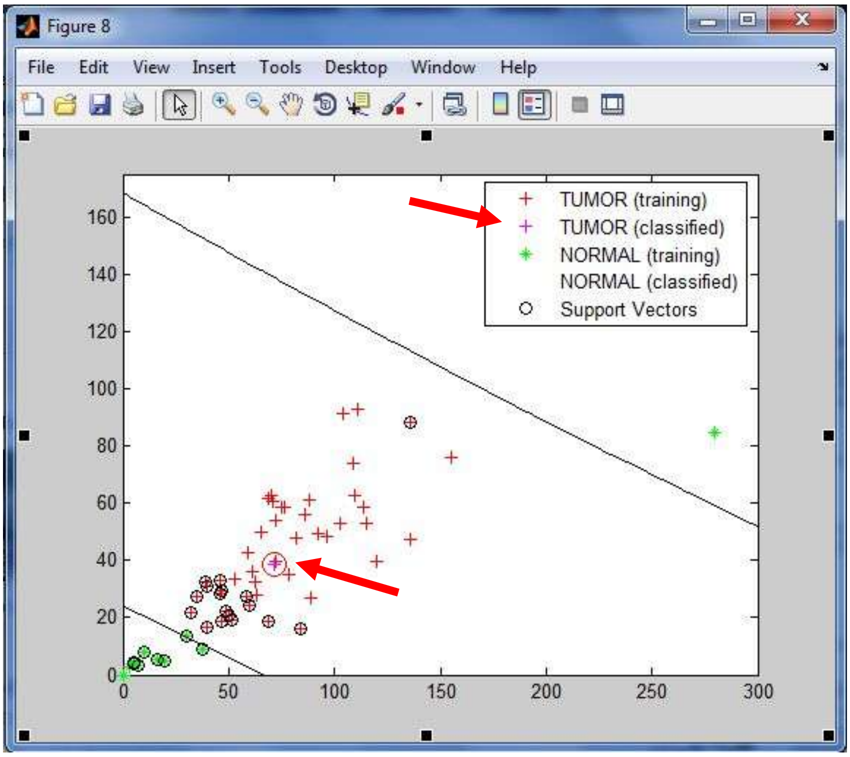


Figure 5.31 Image classification using SVM Test Process, Feature 5 6, Test Image (b)

Test Process with image C:

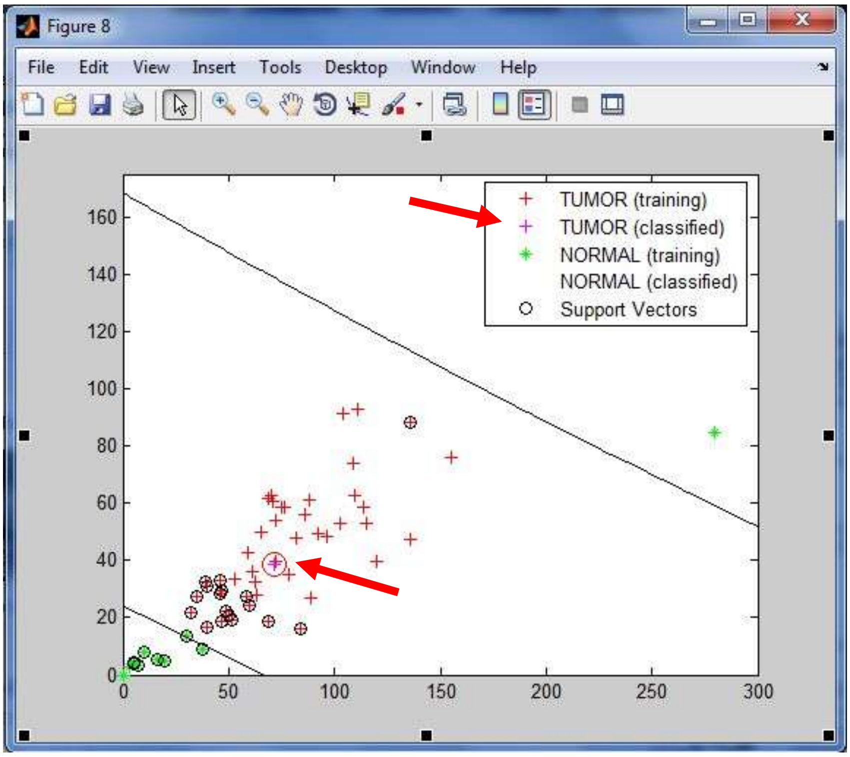


Figure 5.32 Image classification using SVM Test Process, Feature 5 6, Test Image (c)

5.6.2.1.4 Feature 7 and 8's Train and Test Processes with 3 Input Images

Results of training and classification for image are shown in Figure 5.33-5.36.

Classification Result:

- (a) image: image has a "**TUMOR**" and it is an abnormal breast.
- (b) image: image has no tumor and it is a "**NORMAL**" breast.
- (c) image: image has no tumor and it is a "**NORMAL**" breast.

Train Process:

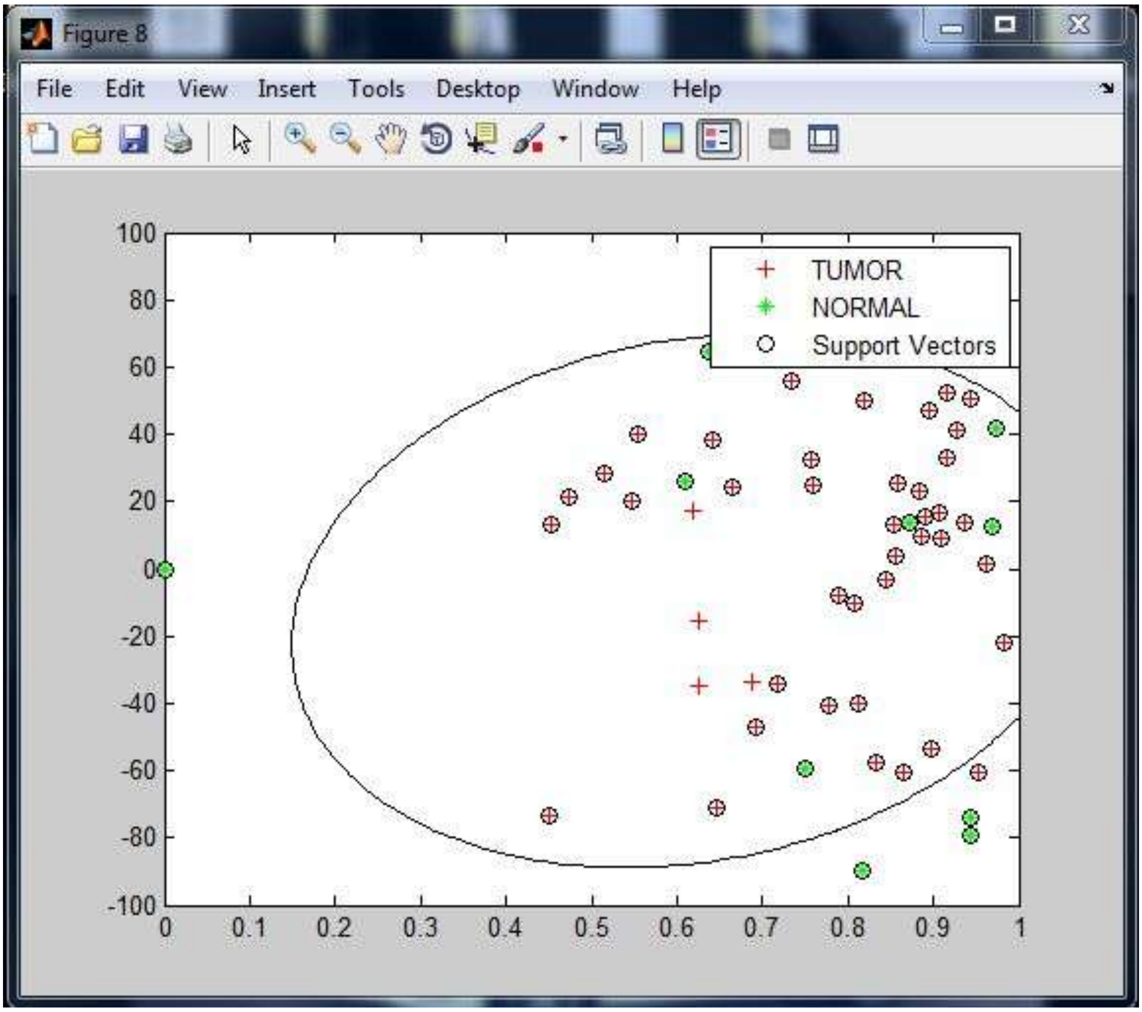


Figure 5.33 Image classification using SVM Train Process, Feature 7 8

Test Process with image A:

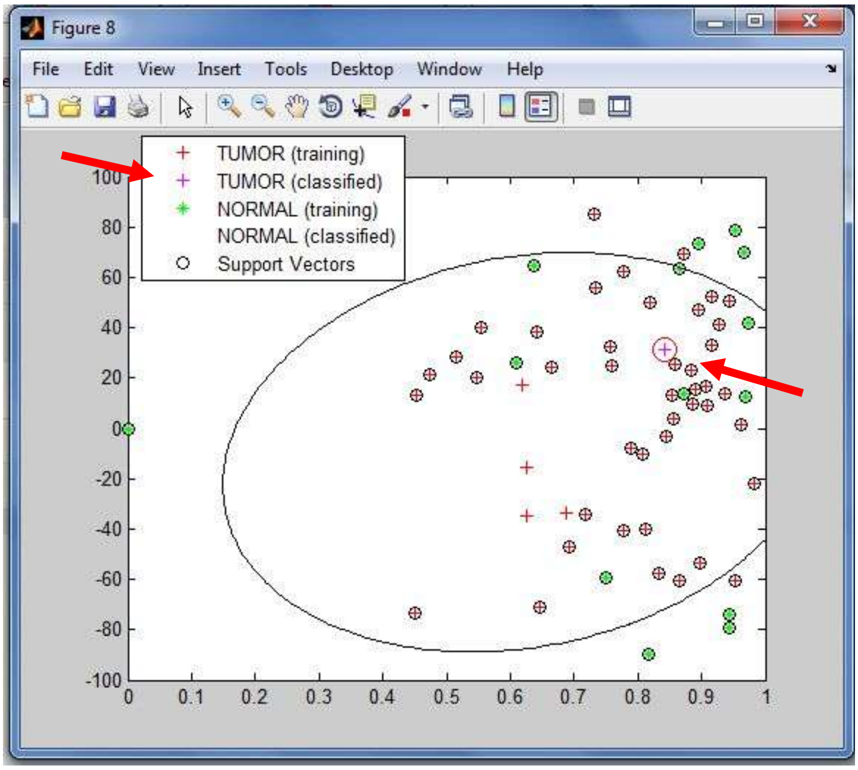


Figure 5.34 Image classification using SVM Test Process, Feature 7 8, Test Image (a)

Test Process with image B:

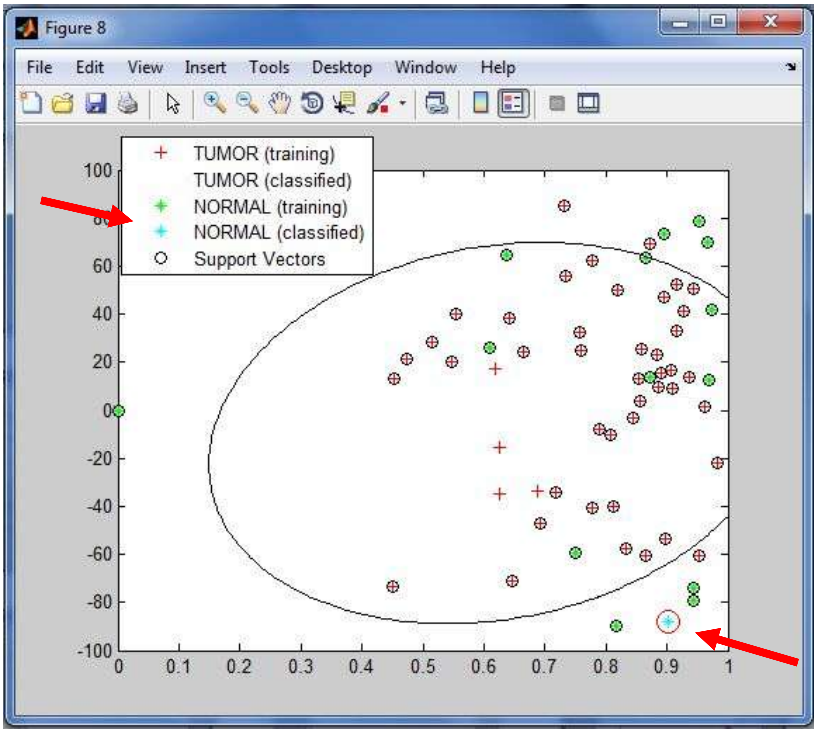


Figure 5.35 Image classification using SVM Test Process, Feature 7 8, Test Image (b)

Test Process with image C:

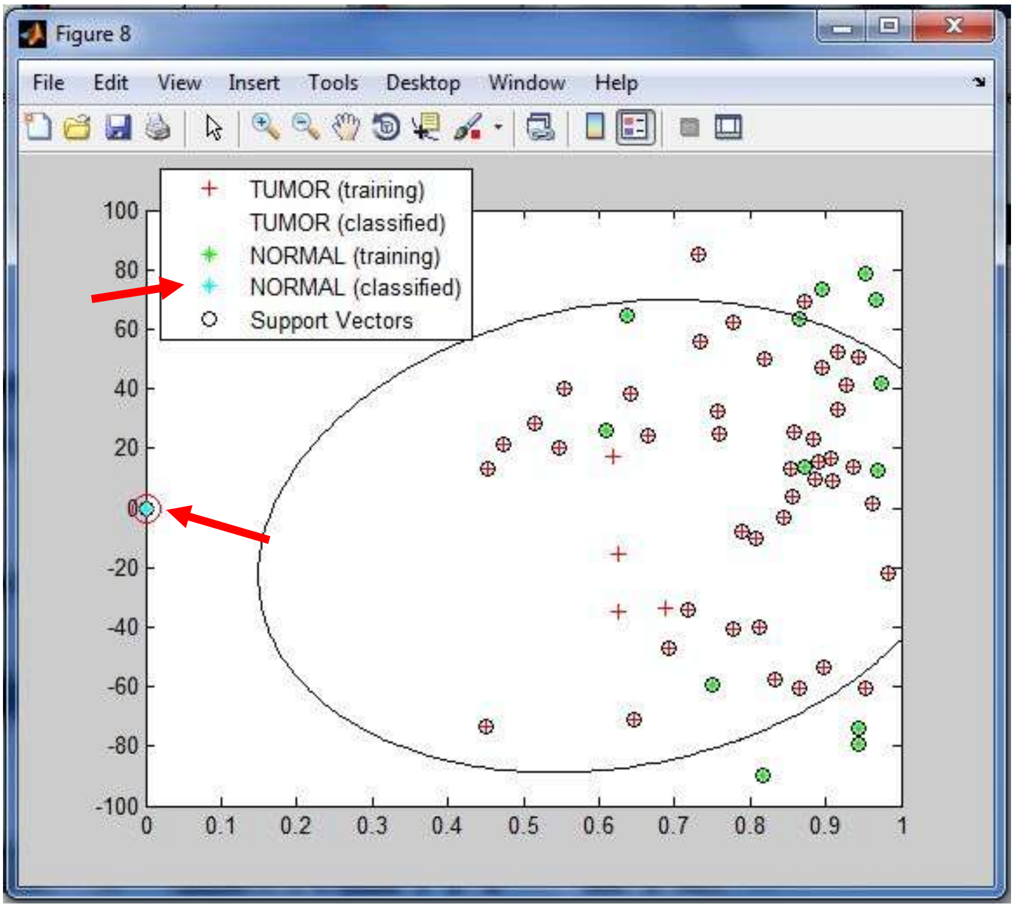


Figure 5.36 Image classification using SVM Test Process, Feature 7 8, Test Image (c)

5.6.2.1.5 Feature 9 and 10's Train and Test Processes with 3 Input Images

Results of training and classification for image are shown in Figure 5.37-5.40.

Classification Result:

- (a) image: image has a "**TUMOR**" and it is an abnormal breast.
- (b) image: image has no tumor and it is a "**NORMAL**" breast.
- (c) image: image has no tumor and it is a "**NORMAL**" breast.

Train Process:

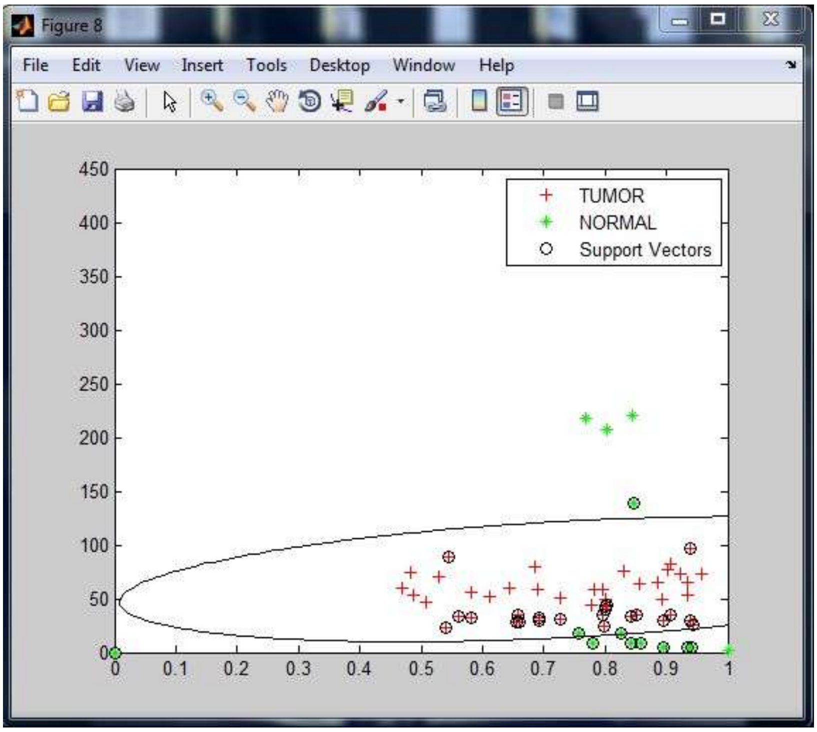


Figure5.37 Image classification using SVM Train Process, Feature 9 10

Test Process with image A:

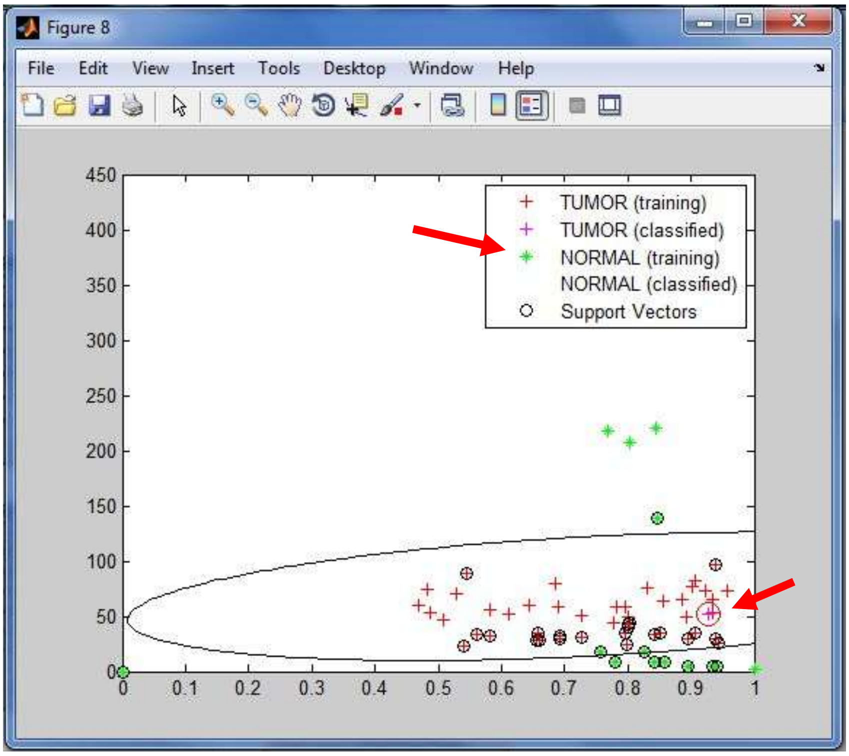


Figure 5.38 Image classification using SVM Test Process, Feature 9 10, Test Image (a)

Test Process with image B:

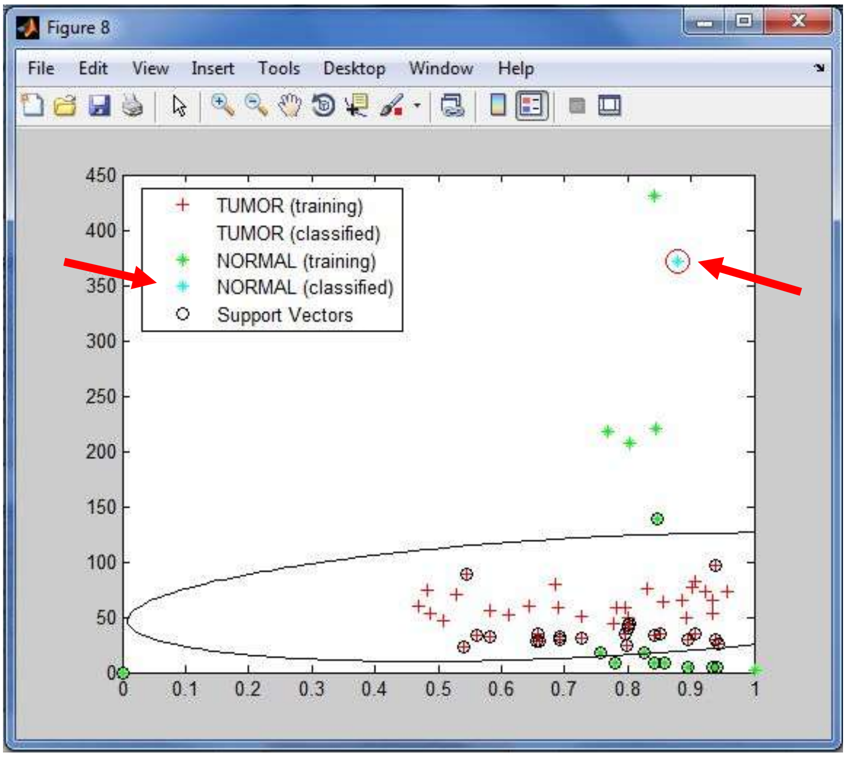


Figure 5.39 Image classification using SVM Test Process, Feature 9 10, Test Image (b)

Test Process with image C:

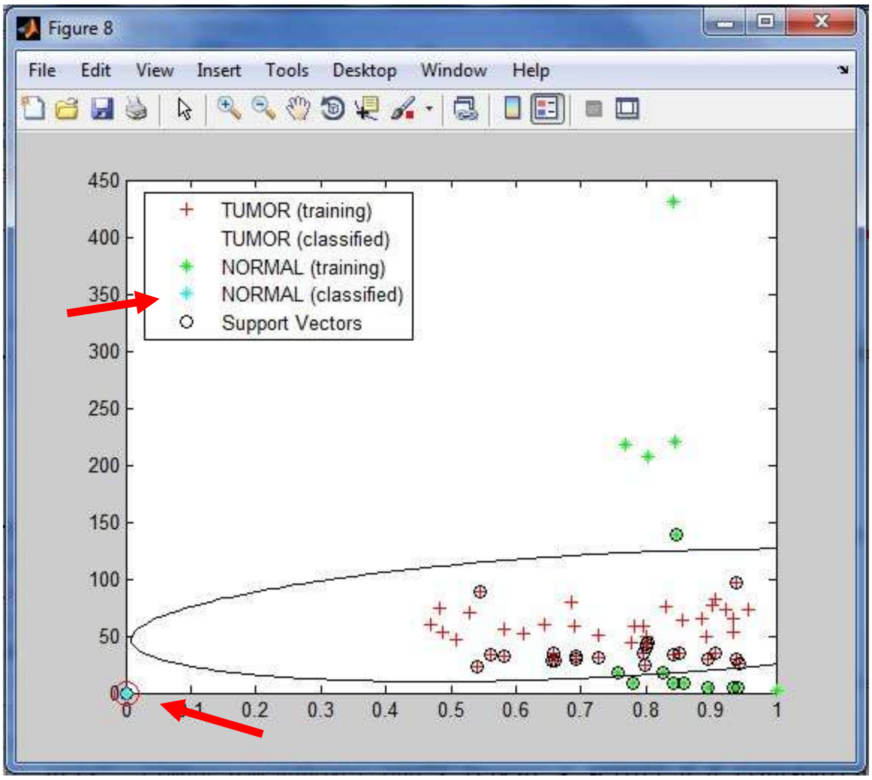


Figure 5.40 Image classification using SVM Test Process, Feature 9 10, Test Image (c)

5.6.2.1.6 Feature 1 to 10's Train and Test Processes with 3 Input Images

Classification Result:

- (a) image: image has a "**TUMOR**" and it is an abnormal breast.
- (b) image: image has no tumor and it is a "**NORMAL**" breast.
- (c) image: image has no tumor and it is a "**NORMAL**" breast.

After train and testing processes for each two feature, test processes have done for 10 features at same time from feature 1 to 10. As a result, developed software has given same results but Matlab cannot have the ability to show graphic that have higher dimensions than 2. Therefore, there is no figure for this process but the results are listed and they are same as previous ones.

5.6.2.2 Section 2: Results of Image Classification with Train and Test Processes

At this section, 92 images used 48 of them for train processes and 44 for test processes. All images are used for classification processes depending on following features.

Features:

i-	Area	(Feature 1)
ii-	Filled Area	(Feature 2)
iii-	Centroid X	(Feature 3)
iv-	Centroid Y	(Feature 4)
v-	Major Axis Length	(Feature 5)
vi-	Minor Axis Length	(Feature 6)
vii-	Eccentricity	(Feature 7)
viii-	Orientation	(Feature 8)
ix-	Solidity	(Feature 9)
x-	EquivDiameter	(Feature 10)

5.6.2.2.1 Feature 1 and 2's Train and Test Processes

Train Process:

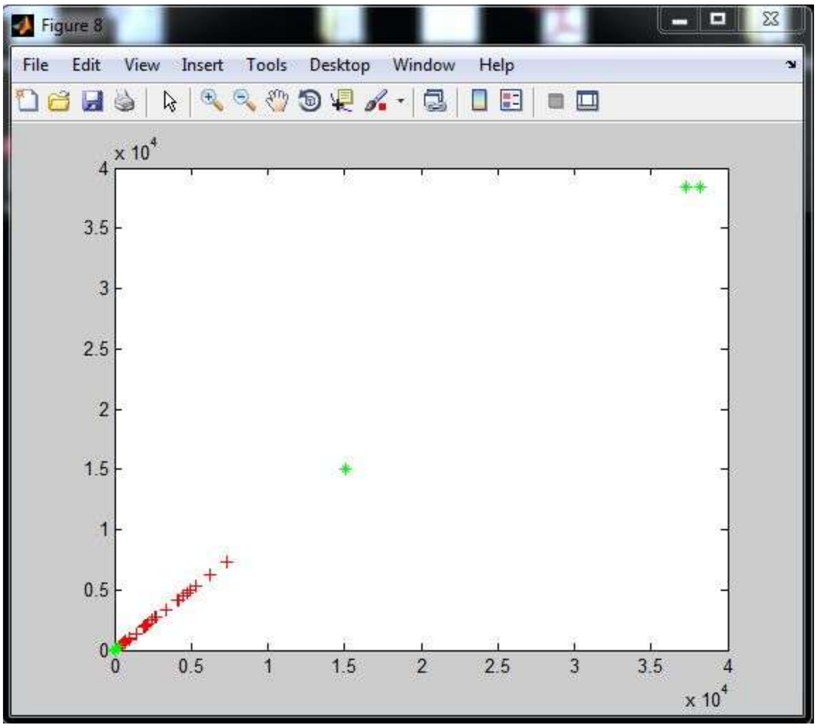


Figure 5.41 Image classification using SVM Train Process, Feature 1 2

Test Process:

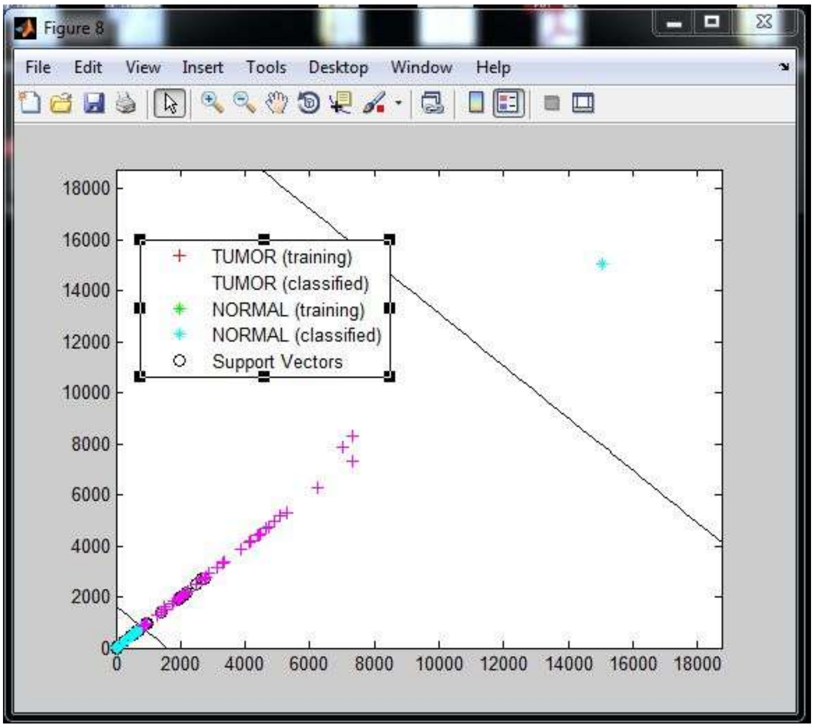


Figure 5.42 Image classification using SVM Test Process, Feature 1 2

5.6.2.2.2 Feature 3 and 4's Train and Test Processes

Train Process:

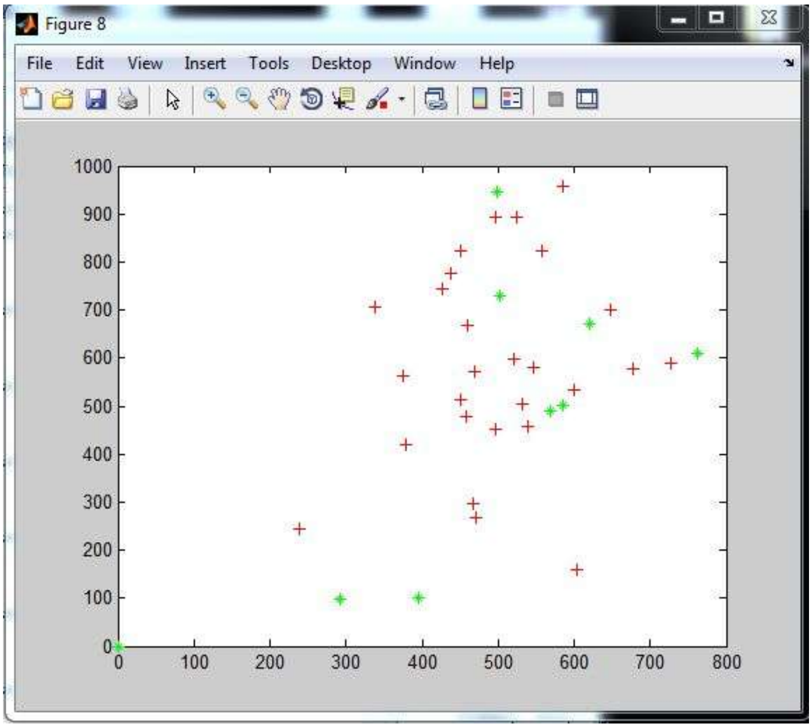


Figure 5.43 Image classification using SVM Train Process, Feature 3 4

Test Process:

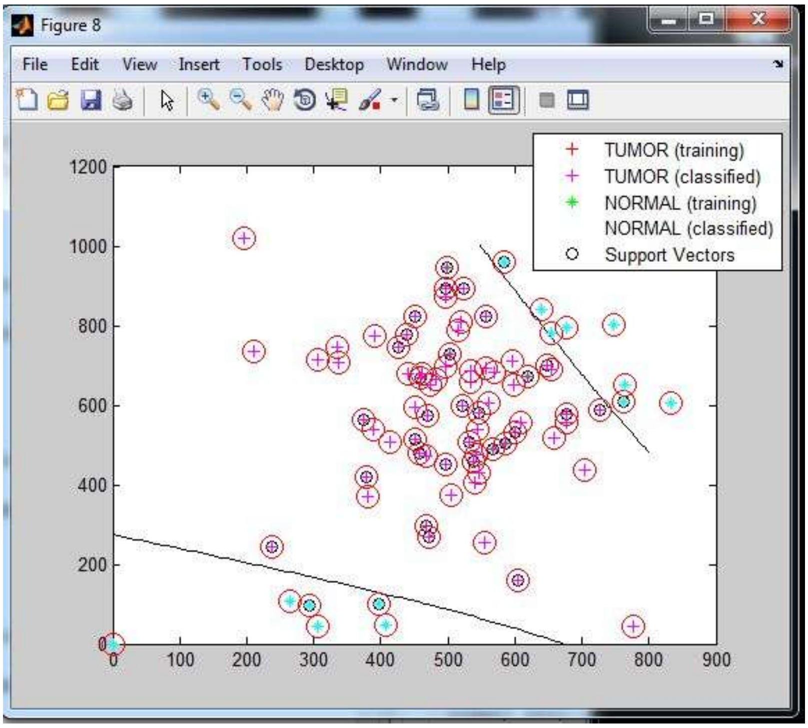


Figure 5.44 Image classification using SVM Test Process, Feature 3 4

5.6.2.2.3 Feature 5 and 6's Train and Test Processes

Train Process:

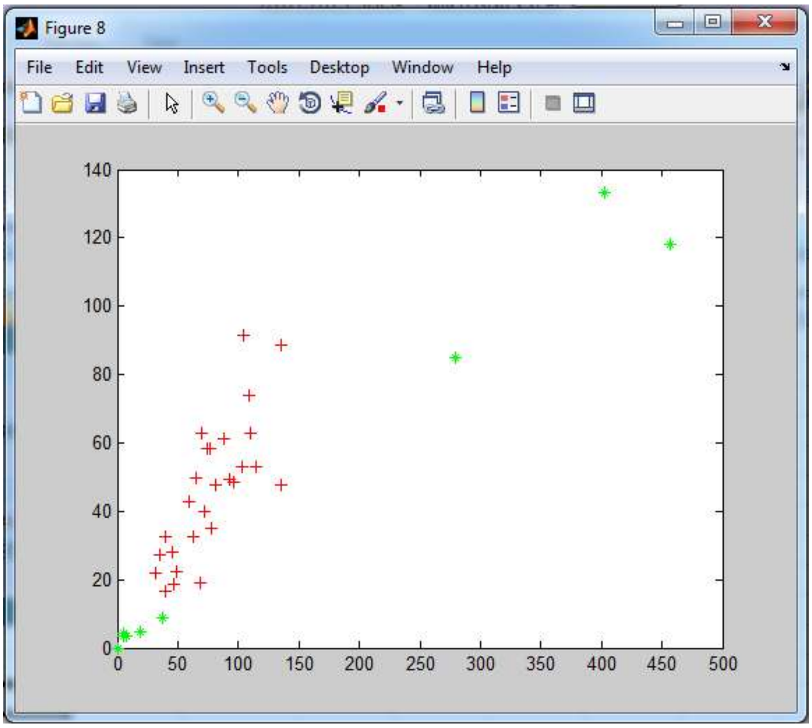


Figure 5.45 Image classification using SVM Train Process, Feature 5 6

Test Process:

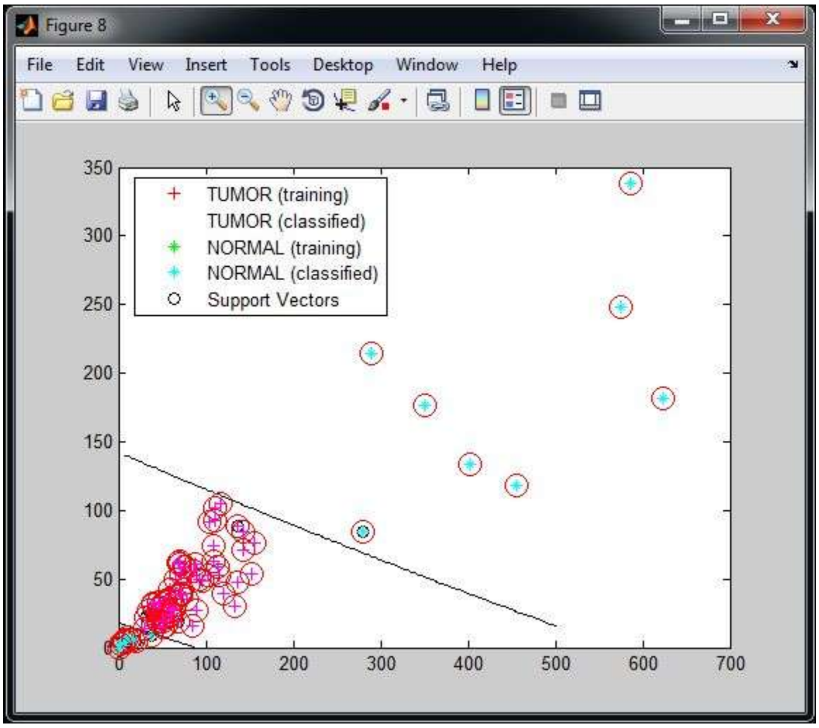


Figure 5.46 Image classification using SVM Test Process, Feature 5 6

5.6.2.2.4 Feature 7 and 8's Train and Test Processes

Train Process:

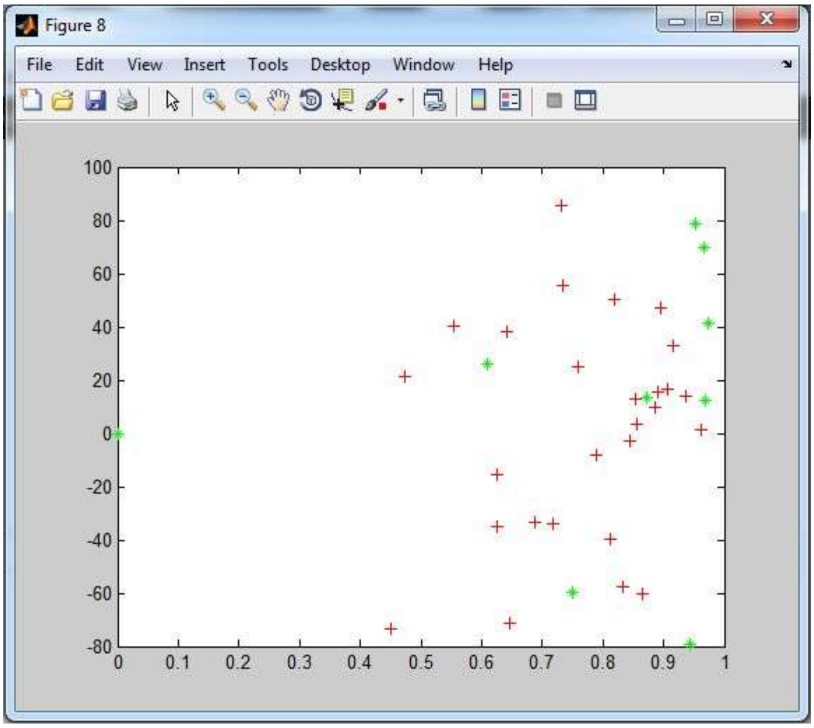


Figure 5.47 Image classification using SVM Train Process, Feature 7 8

Test Process:

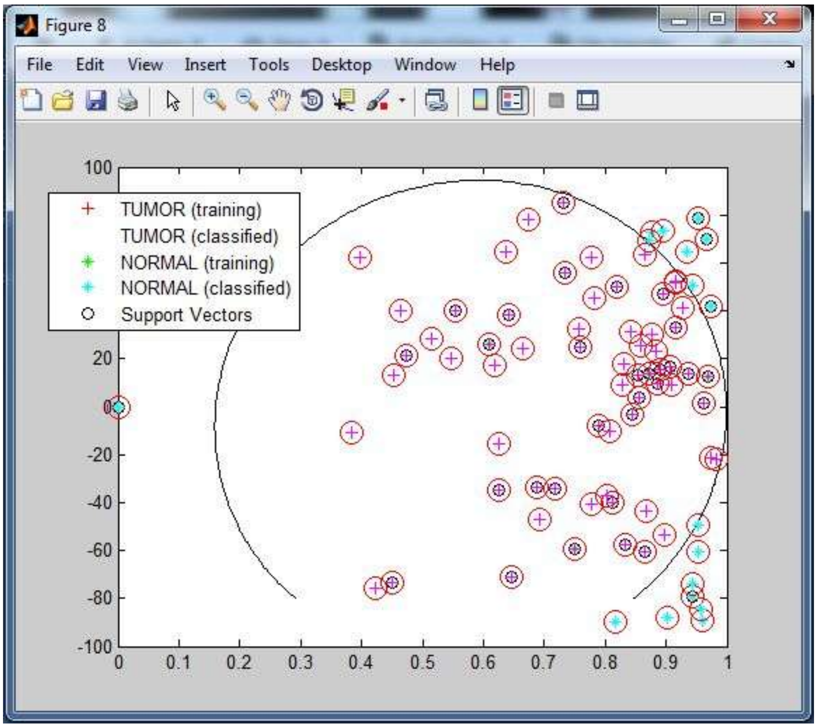


Figure 5.48 Image classification using SVM Test Process, Feature 7 8

5.6.2.2.5 Feature 9 and 10's Train and Test Processes

Train Process:

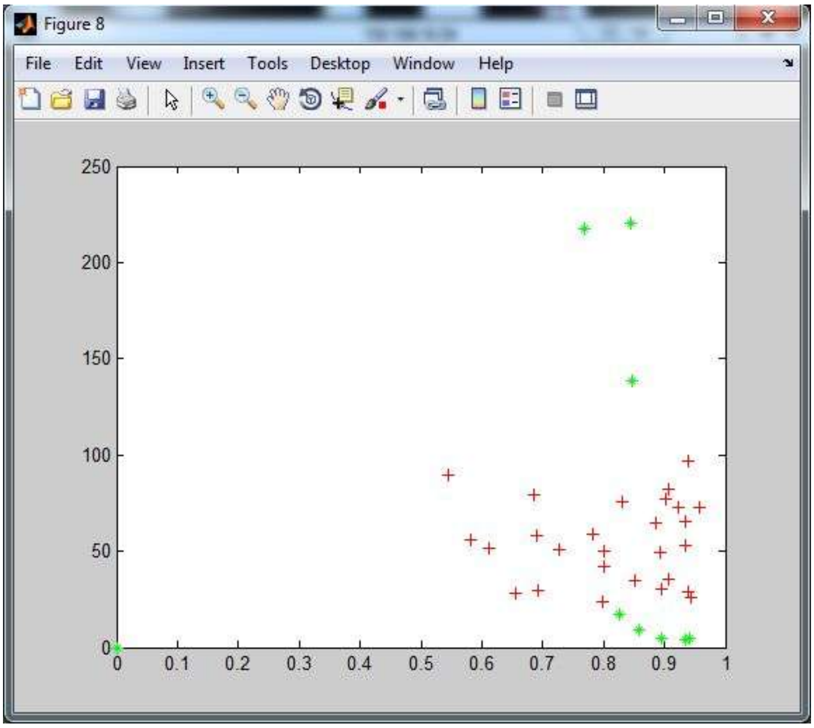


Figure 5.49 Image classification using SVM Test Process, Feature 9 10

Test Process:

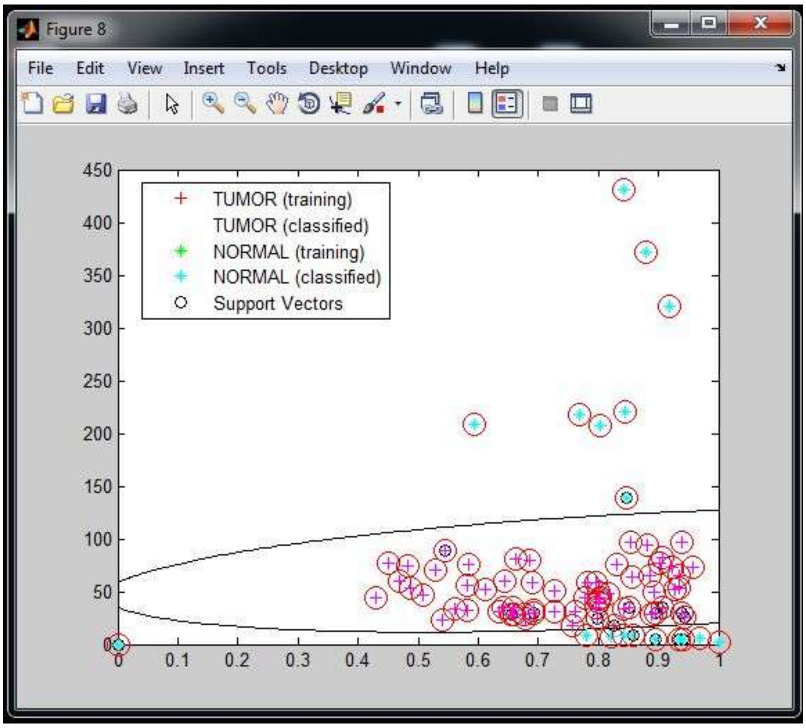


Figure 5.50 Image classification using SVM Test Process, Feature 9 10

5.6.2.3 Accuracy of Classification Results

At this section, results of classification process are listed. All of the gathered information about train and test processes are represented in the following table.

Accuracy of classification process listed below by using 10 features and image classes. Classes are "Tumor" and "Normal" and Features are "Area (1)", "Filled Area (2)", "Centroid X (3)", "Centroid Y (4)","Major Axis Length (5)", "Minor Axis Length (6)", "Eccentricity (7)", "Orientation (8)", "Solidity (9)", "EquivDiameter (10)".

First of all, 92 images taken from MIAS mammographic breast images database and they are separated into two categories which are "Training Dataset Images" and "Test Dataset Images". After that, train and test dataset images categories into two classes "Tumor" and "Normal" depending on images' normal and abnormal tissues. Finally, classification process have done and accuracy has calculated on training dataset, test dataset and overall images. Accuracy percentages are listed below.

	ACCURACY		
Features (1,2,3,4,5,6,7,8,9,10)	Training Dataset	Test Dataset	Overall
Feature 1 and 2	87,50 %	77,30 %	82,40 %
Feature 3 and 4	87,50 %	79,60%	83,55 %
Feature 5 and 6	100,00 %	86,40%	93,20 %
Feature 7 and 8	91,70 %	70,50 %	81,10 %
Feature 9 and 10	97,90 %	86,40 %	92,15 %
Feature 1, 2, 3, 4, and 5	97,90 %	86,40 %	92,15 %
Feature 6, 7, 8, 9, and 10	100 %	93,20 %	96,60 %
Feature 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10	**100 %**	**88,70 %**	**94,30 %**

Table 5.1. Accuracy Rates of SVM Classification of Developed Application

As a result, according to different features SVM classification has different classification accuracy and by increasing features for SVM classification accuracy rates increases. In addition, this application SVM classification at least 81,10% and at most 96,60% at overall images. Application has **94,30 % accuracy** for overall images and all features.

To sum up, at this section, developed software explained with details. All of the techniques that are used while software development listed and explained. In addition, all of the figures of each step are listed and shown including image enhancement, tumor area segmentation, and classification with train and test stages. Furthermore, all the Matlab codes of those steps are listed as well. Therefore, a software was developed that can be detect breast cancer at early stages and after train process, it can decide for each image belong which class depending on the learning algorithms gained while training process. In addition, results of training and test processes listed and accuracy of results are calculated.

CHAPTER 6

CONCLUSION

Image classification is an attempt to label (often textual) an image with appropriate identifiers. These identifiers are determined by the area of interest, whether it is general classification for arbitrary pictures (for instance, from the internet), or a specific domain, for instance, medical x-ray images or geographical images of terrain. Image classification is the process of assigning all pixels in a digital image to particular classes according to their characteristics.

The main task to be accomplished in this project is implementing an image classification system using SVM for early detection of breast cancer and classification purposes. SVM is used for classification because it is very compromising machine learning system and has a good accuracy in results and it can be modified for linear and non-linear classification processes.

In the thesis the types of image classification, their advantages and disadvantages, and also supervised and unsupervised image classification techniques have been described. The importance of medical image classification and their practical implementation have been discussed. The techniques used for medical image classification, steps of image classification have been given. Medical image acquisition using Magnetic Resonance Imaging (MRI), Computer Tomography (CT), Mammography are explained. Image enhancement using stretching, filtering, wavelet transform, feature extraction and classification steps are presented.

The mathematical description of the support vector machines (SVM), linear and nonlinear SVM, the used kernel functions are described. The development of medical image classification system using SVM is performed. Image acquisition, enhancement, feature extraction and classification blocks are presented on breast cancer images. Training and classification results are listed for breast cancer images. Accuracy of developed system is determined. The implementation of image classification system has been done using Matlab package.

To sum up, a software developed that can make early detection of tumor on mammographic breast images by using some sort of techniques like Gaussian smoothing filter, contrast stretching, tophat filtering, discrete wavelet transform, machine learning like Support Vector Machines and image dataset called MIAS mammographic image database. In addition, Matlab software development environment used to develop software and making simulations of each train, test, and classification processes. As a result, developed software worked with 92 images from MIAS database and some sort of test are done and classification results are taken. According to classification results, accuracy of developed software calculated and developed software works with 94,30 % accuracy.

REFERENCES

1. Fisher, R., Perkins, S., Walker, A., & Wolfart, E. (2003). *Image Processing Learning Process Resources, Classification*. Retrieved August 13, 2012, from http://homepages.inf.ed.ac.uk/rbf/HIPR2/classify.htm
2. Natural Resources Canada, (2008). *Image Classification and Analysis*. Retrieved August 13, 2012, from http://www.nrcan.gc.ca/earth-sciences/geography-boundary/remote-sensing/fundamentals/1920
3. Muhammad, H. H. (2006, December 15). *Lecture 13a: Medical Image Analysis* [PowerPoint slides]. Retrieved from http://medim.sth.kth.se/6l2872/F/
4. Abimanyu, N. N., Gugapriya, S., & Lalitha D. (2010, April). Course Project Report: Medical Image Classification using Support Vector Machines. Anna University. Cehnnai, Indian.
5. Digital Image Processing. (2006, September 9). *Lecture 6: Digital Image Processing* [PowerPoint slides]. Retrieved from http://www.scribd.com/doc/91483440/Week-6-Lecture1-2006
6. Grass Tutorial (n.d.). *Classifying Images*. Available from http://grass.fbk.eu/gdp/grass5tutor/HTML_en/c1622.html
7. Image Classification II. (2007, March 6). *Lecture 9: Image Classification II* [PowerPoint slides]. Retrieved from http://www.coursehero.com/file/2717249/Lec9ClassificationII/
8. Tangjaitrong, S. (1999). *Remote Sensing Lecture Notes.* Avaliable from http://www.sc.chula.ac.th/courseware/2309507/Lecture/remote18.htm
9. Khalil, R. (2009, May 29). Lecture LA502: Special Studies Remote Sensing [PowerPoint slides]. Retrieved from http://rmohamed.kau.edu.sa/Files/0052473/files/25518_LA502_07_unsupervised%20classification.pdf
10. Gong, P. (1997). *Conventional Multispectral Classification Methods*. Available from http://nature.berkeley.edu/~penggong/textbook/chapter7/html/sect73.htm
11. Bryant, R. G. (1999, October 11). *Lecture 12: Introduction to Remote Sensing* [PowerPoint slides]. Retrieved from http://www.slideserve.com/loretta/introduction-to-remote-sensing-lecture-12

12. McGinty, C. (2006, November 06). *Lecture: Supervised Classification Part1* [PowerPoint slides]. Retrieved from http://www.gis.usu.edu/~doug/RS5750/assign/lab10.pdf
13. Muhammad, H. H. (2006, December 15). *Lecture 13b: Medical Image Analysis* [PowerPoint slides]. Retrieved from http://medim.sth.kth.se/6l2872/F/
14. Image Classification Types. (n. d.). Available from http://www2.geog.soton.ac.uk/users/TrevesR/obs/rseo/image_classification_types.html
15. Smitha P., Shaji L., & Mini, Dr. M. G. (2011). A review of medical image classification techniques. *International Journal of Computer Applications® (IJCA): Communication & Instrumentation*, p 34-38.
16. Liew, Dr. S. C. (2001). CRISP *Tutorial: Image Processing and Analysis*. Retrieved from http://www.crisp.nus.edu.sg/~research/tutorial/process.htm
17. Rosenfeld, A. (2005). Image Pattern Recognition. *General Topics for Engineers (Math, Science & Engineering)* , 69 (5), 596 - 605. doi:10.1109/PROC.1981.12027
18. Patih, J. K., & Kumor, R. (2011). Advances in image processing for detection of plant diseases. *Journal of Advanced Bioinformatics: Applications and Research*, 2(2), 135-141.
19. Antonie, M., Zaiane O. R., & Coman, A. (2001). Application of Data Mining Techniques for Medical Image Classification. *Conference of ACM: Proceedings of the Second International Workshop on Multimedia Data Mining*, 94-101.
20. Seibert, J. A. (2012). *Medical Image Data Characteristics.* Avaliable from http://www.siimcenter.org/books/archiving/chapter-2-medical-image-data-characteristics-capicity-requirements
21. Tomography. (2012). *Invert Website: Tomography Imaging*. Retrieved March 03, 2013, from http://www.bath.ac.uk/elec-eng/invert/tomography.html
22. Mammography. (2013). Radiological Society of North America Website: Mammography. Retrieved March 03, 2013, from http://www.radiologyinfo.org/en/info.cfm?pg=mammo
23. Mueller, K. (2007, October 06). *Lecture 5: Image Operations (Introduction to Medical Imaging)* [PowerPoint slides]. Retrieved from http://www.cs.sunysb.edu/~mueller/teaching/cse377/imageOps.pdf
24. Image Enhancement Techniques (2012). [PowerPoint slides]. Retrieved from http://seminarprojects.com/Thread-image-enhancement-techniques-full-report

25. Gruber, T. (2001). *Contrast Enhancement*. Retrieved from http://www.fastgraph.com/help/contrast_enhancement.html

26. Fiete, R. D. (2010). Image Enhancement Processing. *Society of Photo-Optical Instrumentation Engineers*, 127-132. doi:10.1117/3.868276.ch9

27. The GIMP Documentation Team. (2012). *Stretch Contrast.* Retrieved September 02, 2012, from http://docs.gimp.org/en/plug-in-c-astretch.html

28. Kolas, O. (2005). *Image processing with gluas.* Retrieved September 02, 2012, from http://codecave.org/image-processing/chapter-automaticadjustments.html

29. Jain, R., Kasturi, R., & Schunck, B. G. (1995). *Machine Vision.* United States of America: McGraw-Hill.

30. Petrakis, E. G. M. (2003, October 15). *Computer Vision Course: Filtering* [PowerPoint slides]. Retrieved from http://www.intelligence.tuc.gr/~petrakis/courses/computervision/filtering.pdf

31. RoboRealm. (2005). *Vision for Machines: Min filtering*. Retrieved from http://www.roborealm.com/help/Min.php

32. RoboRealm. (2005). *Vision for Machines: Max filtering*. Retrieved from http://www.roborealm.com/help/Max.php

33. Vandevenne, L. (2004). *Lode's Computer Graphics Tutorial: Image Filtering*. Retrieved from http://lodev.org/cgtutor/filtering.html

34. Astrophoto, P. (2010). *Minimum and Maximum Filters*. Retrieved from http://pixinsight.com/doc/legacy/LE/19_morphological/minmax_filter/minmax_filter.html

35. Schulze, M. A. (2001). *What are the Mean and Median Filters*. Retrieved from http://www.markschulze.net/java/meanmed.html

36. Median Filter. (2007, August 16). *Wikipedia: The free encyclopedia.* Retrieved September 05, 2012, from http://en.wikipedia.org/wiki/File:Median_filter_example.jpg

37. Nixon, M., & Aguado, A. (2002). *Gaussian Filtering*. Retrieved from http://users.ecs.soton.ac.uk/msn/book/new_demo/gaussian/

38. Mathworks (2012). *Matlab Documentation: Top-Hat Filtering*. Retrieved from http://www.mathworks.com/help/images/ref/imtophat.html

39. Mathworks (2012). *Matlab Documentation: Image Transform*. Retrieved from http://www.mathworks.com/discovery/image-transform.html

40. Mathworks (2012). *Matlab Documentation: Wavelet Transforms*. Retrieved from http://www.mathworks.com/discovery/wavelet-transforms.html

41. Addison, P. S. (2005). Image Wavelet Transforms and the ECG: a review. *Institute of Physics Publishing*, Physiol. Meas. 26 (2005) R155-199. doi: 10.1088/0967-3334/26/5/R01
42. Bruce, A., Donoho, D., & Gao, H. Y. (1996). Wavelet Analysis [for signal processing]. *IEEE Spectrum*, 33(2), 26-35.
43. Lester, E. D., 1998. FEATURE EXTRACTION, IMAGE SEGMENTATION, AND SURFACE FITTING: *THE DEVELOPMENT OF A 3D SCENE ECONSTRUCTION SYSTEM* [Master Thesis].
44. Feature Selection. (2012, September 21). *Wikipedia: The free encyclopedia.* Retrieved September 09, 2012, from http://en.wikipedia.org/wiki/Feature_selection
45. Hulle M. V., & Davis, J. (2009, April 04). *Data Mining Lecture: Feature Extraction & Feature Selection* [PowerPoint slides]. Retrieved from http://134.58.34.50/~marc/DM_course/slides_selection.pdf
46. Seetha, M., MuraliKrishna, I. V., & Deekshatulu, B. L. (n.d.). *Comparison of Advanced Techniques of Image Classification.* Retrieved September 13, 2012, from http://www.gisdevelopment.net/proceedings/mapworldforum/rsp/MWF_Remote_78.pdf
47. Dobrescu, R., Dobrescu, M., Mocanu, S., & Popescu, D. (2010). Medical image classification for skin cancer diagnosis based on combined texture and fractal analysis. *WSEAS Transactions on Biology and Biomedicine*, 7(3), 223-232.
48. Smitha P., Shaji L., & Mini, Dr. M. G. (2011). A review of medical image classification techniques. *International Journal of Computer Applications® (IJCA): Communication & Instrumentation*, p 34-38.
49. Noble, W. S., (2006). What is a Support Vector Machine? *Nature Biotechnology,* 24(12), 1565-1567.
50. Mathworks (2012). *Matlab Documentation: Continuous Wavelet Transform*. Retrieved from http://www.mathworks.com/help/wavelet/gs/continuous-wavelet-transform.html
51. Sripathi, D. (2003). *Thesis: Discrete Wavelet Transform*. Retrieved from http://etd.lib.fsu.edu/theses/available/etd-11242003-185039/unrestricted/09_ds_chapter2.pdf
52. Klapetek, P., Necas, D., & Anderson, C. (2012). *Gwyddion user guide: Wavelet Transform*. Retrieved from http://gwyddion.net/documentation/user-guide-en/wavelet-transform.html

53. Discrete Wavelet Transform. (2012, August 26). *Wikipedia: The free encyclopedia.* Retrieved September 10, 2012, from http://en.wikipedia.org/wiki/Discrete_wavelet_transform

54. Rios, D. (2010). *Neuro AI – Artificial Intelligence and Neural Networks.* Retrieved from http://gwyddion.net/documentation/user-guide-en/wavelet-transform.html

55. Genetic Algorithms (n.d.). Retrieved from http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/tcw2/report.html#Introduction

56. StatSoft, Inc. (2012). Electronic Statistics Textbook. Tulsa, OK: StatSoft. WEB: http://www.statsoft.com/textbook/.

57. Srivastava, D. K., & Bhambhu, L. (2009). Data classification using support vector machine. *Journal of Theoretical and Applied Information Technology,* p 1-5.

58. Souza, C. (2012). *Kernel Functions for Machine Learning Applications.* Retrieved from http://crsouza.blogspot.com/2010/03/kernel-functions-for-machine-learning.html

59. Ben-Hur, A., & Weston, J. (2006). *A User's guide to support vector machines.* p 1-18.

60. Maiga, A. (2011, March 05). *An introduction of kernel methods for support vector machines* [PowerPoint slides]. Retrieved from http://www.ptidej.net/seminar/110218%20-%20SVM%20Kernels/110306%20-%20Kernel%20Method%20-%20SVM.pdf

61. Campbell, C., Frieb, T. T., & Cristianini, N. (n.d.). *Maximal Margin Classification using the KA Algorithm p. 3-4.*

62. Olkkonen H., 2011. *Discrete Wavelet Transforms - Biomedical Applications p. 28*

63. Hostalkova, E., & Prochazva, A., 2007. COMPLEX WAVELET TRANSFORM IN BIOMEDICAL IMAGE DENOISING p. 1

64. Guyon, I., & Elisseeff, A., 2006. An introduction to Feature Extraction p. 3

65. Microsoft (2013). *Microsoft Research: The Standard SVM Formulation*. Retrieved from http://research.microsoft.com/en-us/um/people/manik/projects/trade-off/svm.html

66. Girolami, M., Mischak, H., & Krebs, R. (2006). Analysis of complex, multidimensional datasets. *Drug Discovery Today: Technologies*, 3(1), 13-19.

APPENDICES

Appendix 1 - Matlab Code of Developed Software; Train Process

```
%% Breast Cancer Classification System -- Train Phase
%% Selection of Image for Train Process

cd('c:\Program Files\MATLAB\R2011b\work\Traindata\');
imagelist = dir('*.pgm');
num = numel(imagelist);
imdata = cell(1,numel(imagelist));

%% Making List of Image in the Train Folder
for k=1:num imdata{k} = imread(imagelist(k).name); end

%% Declaration of Feature Space Variables for Segmented Area %%%%%%%
SImgAreaTT(num,1)=zeros;
SImgCentroidXTT(num,1)=zeros;
SImgCentroidYTT(num,1)=zeros;
SImgMajorAxisLengthTT(num,1)=zeros;
SImgMinorAxisLengthTT(num,1)=zeros;
SImgEccentricityTT(num,1)=zeros;
SImgOrientationTT(num,1)=zeros;
SImgFilledAreaTT(num,1)=zeros;
SImgSolidityTT(num,1)=zeros;
SImgEquivDiameterTT(num,1)=zeros;

%% Image Enhacement Process
%% Orijinal Image

for m=1:num,
disp(m);
ImportedImg = imdata{m};

ImgTT = ImportedImg;
%figure(1);imshow(ImgTT,'InitialMagnification',50);title('Original Image');
%pause(0.5);
%close();

%% Converted to Binary (Black&White) Image of Orijinal Image
ImgTTBWU = im2bw(ImgTT,0.70);
%figure(2);imshow(ImgTTBWU,'InitialMagnification',50);title('Binary Version of
Original Image');
%pause(0.5);
%close();

%% Converted to Binary (Black&White) Image and Selected Big Areas Image %%%
ImgBWUUSeg = bwareaopen(ImgTTBWU, 10000);
%figure(3);imshow(ImgBWUUSeg,'InitialMagnification',50);title('Binary Version Areas
of Original Image');
```

```
%pause(0.5);
%close();

%% Binary (Black&White) Image Selected Area is Filled
%%%%%%%%%%%%%%%%%%%%%%%%
ImgBWSegUUFil= imfill(ImgBWUUSeg,'holes');
%figure(4);imshow(ImgBWSegUUFil,'InitialMagnification',50);title('Filled up Areas of
B/W Image');
%pause(0.5);
%close();

%% Binary Image Selected Area Region Properties, BoundingBox Properties %%%
ImgStatsTr = regionprops(ImgBWSegUUFil, 'BoundingBox'); %BoundingBox Prop
ImgStatSizeTr=size(ImgStatsTr);                 %List of Output Stats
ImgStatSizeTrX=ImgStatSizeTr(1);                %No of Picked Areas

for ii=1:ImgStatSizeTrX                          %Chosing right
        BoxList{ii}=ImgStatsTr(ii).BoundingBox;     %BoundingBox
        if((ImgStatsTr(ii).BoundingBox(2))<10)      %BoundingBox 2nd value
           selectedBoxTrIdx=ii;                     %If it is <10, OK.
           selectedBoxTr=BoxList{ii};
        end
end

%% Crop BoundingBoxed Image From Original Image
ImgUUCrop= imcrop(ImgTT,ImgStatsTr(selectedBoxTrIdx).BoundingBox);
%figure(5);imshow(ImgUUCrop,'InitialMagnification',50);title('Cropped Region of
Original Image');
%pause(0.5);
%close();

%% Declaration of Variables for Removing BoundingBoxed Region/Area x=real(zeros);
y=real(zeros);
xl=real(zeros);
yl=real(zeros);
xp=real(zeros);
yp=real(zeros);
%%%%%%%%%%%%%%%%%%%%%%%%%%
x=(round(ImgStatsTr(ii).BoundingBox(1)))-30;
y=(round(ImgStatsTr(ii).BoundingBox(2)));
xl=x+(round(ImgStatsTr(ii).BoundingBox(3)))+30;
yl=y+(round(ImgStatsTr(ii).BoundingBox(4)))+150;
xp=x+1; yp=y+10;
xloop=xl-x;
yloop=(yl-y)/(yp-y);
%%%%%%%%%%%%%%%%%%%%%%%%%%
% Removes BoundingBoxed Region/Area On the Original Image Pixels -> Black %
if((round(ImgStatsTr(ii).BoundingBox(1)))>300)
          for jj=1:yloop
```

```
            ImgTT(y:yp,xp:xl)=0;
            xp=xp+4;
                    y=y+10;
                    yp=yp+10;
    end
else
            for j=1:yloop
        ImgTT(y:yp,xp:xl)=0;
        xl=xl-4;
                y=y+10;
                yp=yp+10;
            end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%
x=real(zeros);
y=real(zeros);
xl=real(zeros);
yl=real(zeros);
xp=real(zeros);
yp=real(zeros);
xloop=real(zeros);
yloop=real(zeros);
%%%%%%%%%%%%%%%%%%%%%%%%%
% Applying Gaussian Smoothing Filter on the modified Image
HTTr = fspecial('gaussian',[5 5],1);
ImgGFTT = imfilter(ImgTT,HTTr,'replicate','same');
% figure(6); imshow(ImgGFTT,'InitialMagnification',50);title('Gaussian Filtered Image');
% pause(0.5);
% close();

% Applying Contrast Streching on the modified Image
%%%%%%%%%%%%%%%%%%%%%%%%%%
ImgCSTT = imadjust(ImgGFTT,[0.45 1],[]); %stretchlim(ImgGF)
% figure(7), imshow(ImgCSTT,'InitialMagnification',50), title('Contrast Streching of
Image');
% pause(0.5);
% close();

% Applying Top Hat Filtering on the modified Image
seTT = strel('disk',250);
ImgTHFTT = imtophat(ImgCSTT,seTT);
% figure(8), imshow(ImgTHFTT,'InitialMagnification',50), title('Tophat filtering of
Image');
% pause(0.5);
% close();

% Applying Wavelet Filtering on the modified Image
% Two Scale 2D Discrete Wavelet Transform Daubechies, Decomp&Reconst %%%%%%
% Load Filtered Image for Wavelet Transform Decomp&Reconst
```

```
Xtest=ImgTHFTT;
%% Set the level for decomposition.Compute 2-level decomposition of img %%%
N=2;
[C,S] = wavedec2(Xtest,N,'db1');
%%%% Extract the level 1 coefficients
cA1 = appcoef2(C,S,'db1',1);
[cH1,cV1,cD1] = detcoef2('all',C,S,1);
%%%% Extract the level 2 coefficients
cA2 = appcoef2(C,S,'db1',2);
[cH2,cV2,cD2] = detcoef2('all',C,S,2);
%%%% Here are reconstructed branches
A1 = wrcoef2('a',C,S,'db1',1);
A2 = wrcoef2('a',C,S,'db1',2);
H1 = wrcoef2('h',C,S,'db1',1);
V1 = wrcoef2('v',C,S,'db1',1);
D1 = wrcoef2('d',C,S,'db1',1);
H2 = wrcoef2('h',C,S,'db1',2);
V2 = wrcoef2('v',C,S,'db1',2);
D2 = wrcoef2('d',C,S,'db1',2);
%%%% Set the threshold
[thr,sorh,keepapp] = ddencmp('cmp','wv',Xtest);
[XC,CXC,LXC,PERF0,PERFL2]=wdencmp('gbl',C,S,'db1',N,thr,sorh,keepapp);
%%%% Multilevel 2-D wavelet reconstruction
M=waverec2(CXC,LXC,'db1');
out1tr=uint8(M);
%%%% calculate the peak signal to noise ratio
input_ima1=double(Xtest);
    out2=double(out1tr);
    error=0;
     for y=1:191
      for x=1:159
        MSE=((input_ima1(x,y))-(out2(x,y)))^2;
          error=MSE+error;
      end
     end
MSE_WO=(1/(159*191))*error;
PSNR_WO=20*log10(255/sqrt(MSE_WO));
%%%%subtracting the original and reconstructed image
diff_ima=imsubtract(input_ima1,out2);
%figure(9); imshow(out1tr,'InitialMagnification',50); title('Image After Discrete Wavelet
Transform');
%pause(0.5);
%close();

% Segmentation Process after Filtering Image
ImgBWTTr=im2bw(out1tr);
HSTTr = fspecial('gaussian',[5 5],1);
ImgGFBWTTr = imfilter(ImgBWTTr,HSTTr,'replicate','same');
```

```
% figure(10); imshow(ImgGFBWTTr,'InitialMagnification',50);title('Gaussian Filtered and
Segmented Image');
% pause(1);
% close();

% Classification Process after Filtering Image
BWLabelTTr = bwlabel(ImgGFBWTTr);
stats = regionprops(BWLabelTTr,'all');

ImgTrStatSize=size(stats);
ImgTrStatSizeX=ImgTrStatSize(1);

for iii=1:ImgTrStatSizeX
   ImgStatTrAreaList{iii}=stats(iii).Area;
end

if(ImgTrStatSizeX==0)

        SImgAreaTT(k,1)=zeros; %Segmented Image -Area Property
        SImgCentroidXTT(k,1)=zeros;
        SImgCentroidYTT(k,1)=zeros;
        SImgMajorAxisLengthTT(k,1)=zeros;
        SImgMinorAxisLengthTT(k,1)=zeros;
        SImgEccentricityTT(k,1)=zeros;
        SImgOrientationTT(k,1)=zeros;
        SImgFilledAreaTT(k,1)=zeros;
        SImgSolidityTT(k,1)=zeros;
        SImgEquivDiameterTT(k,1)=zeros;

         disp('Breast is normal, there is no TUMOR.');

else

ImgStatAreaLstMatTr=cell2mat(ImgStatTrAreaList);
StatAreaMaxValueTr=max(ImgStatAreaLstMatTr);

if(StatAreaMaxValueTr<300)

        SImgAreaTT(k,1)=zeros; %Segmented Image -Area Property
        SImgCentroidXTT(k,1)=zeros;
        SImgCentroidYTT(k,1)=zeros;
        SImgMajorAxisLengthTT(k,1)=zeros;
        SImgMinorAxisLengthTT(k,1)=zeros;
        SImgEccentricityTT(k,1)=zeros;
        SImgOrientationTT(k,1)=zeros;
        SImgFilledAreaTT(k,1)=zeros;
        SImgSolidityTT(k,1)=zeros;
        SImgEquivDiameterTT(k,1)=zeros;
```

```
    disp('Breast is normal, there is no TUMOR.');
else

if isempty(stats)

    SImgAreaTT(k,1)=zeros; %Segmented Image -Area Property
    SImgCentroidXTT(k,1)=zeros;
    SImgCentroidYTT(k,1)=zeros;
    SImgMajorAxisLengthTT(k,1)=zeros;
    SImgMinorAxisLengthTT(k,1)=zeros;
    SImgEccentricityTT(k,1)=zeros;
    SImgOrientationTT(k,1)=zeros;
    SImgFilledAreaTT(k,1)=zeros;
    SImgSolidityTT(k,1)=zeros;
    SImgEquivDiameterTT(k,1)=zeros;

    disp('Breast is normal, there is no TUMOR.');

else
    [obj1cnt,obj2cnt] = size(stats);
    for stscnt = 1:(obj1cnt)
       arealist(stscnt)=stats(stscnt).Area;
    end
    MaxArea = max(arealist);

range=300:7500;
selectedvalue=max(arealist(arealist>=min(range) & arealist<=max(range)));

if isempty(selectedvalue)
   range=7501:800000;
   selectedvalue=max(arealist(arealist>=min(range) & arealist<=max(range)));
end

if isempty(selectedvalue)

   range=0:300;
   selectedvalue=max(arealist(arealist>=min(range) & arealist<=max(range)));

    SImgAreaTT(k,1)=zeros; %Segmented Image -Area Property
    SImgCentroidXTT(k,1)=zeros;
    SImgCentroidYTT(k,1)=zeros;
    SImgMajorAxisLengthTT(k,1)=zeros;
    SImgMinorAxisLengthTT(k,1)=zeros;
    SImgEccentricityTT(k,1)=zeros;
    SImgOrientationTT(k,1)=zeros;
    SImgFilledAreaTT(k,1)=zeros;
    SImgSolidityTT(k,1)=zeros;
    SImgEquivDiameterTT(k,1)=zeros;
```

```
        disp('Breast is normal, there is no TUMOR.');

end

    index=find(arealist==selectedvalue);
    index=index(1);

    Area=stats(index).Area;
    SImgAreaTT(m,1)=Area;

    CentroidX=stats(index).Centroid(1);
    SImgCentroidXTT(m,1)=CentroidX;

    CentroidY=stats(index).Centroid(2);
    SImgCentroidYTT(m,1)=CentroidY;

    MajorAxisLength=stats(index).MajorAxisLength;
    SImgMajorAxisLengthTT(m,1)=MajorAxisLength;

    MinorAxisLength=stats(index).MinorAxisLength;
    SImgMinorAxisLengthTT(m,1)=MinorAxisLength;

    Eccentricity=stats(index).Eccentricity;
    SImgEccentricityTT(m,1)=Eccentricity;

    Orientation=stats(index).Orientation;
    SImgOrientationTT(m,1)=Orientation;

    FilledArea=stats(index).FilledArea;
    SImgFilledAreaTT(m,1)=FilledArea;

    Solidity=stats(index).Solidity;
    SImgSolidityTT(m,1)=Solidity;

    EquivDiameter=stats(index).EquivDiameter;
    SImgEquivDiameterTT(m,1)=EquivDiameter;

        if (Area>=300 && Area <=7500)
        BWLabelTTSelected = ismember(BWLabelTTr, index);
                            disp('Breast is not normal, there is a TUMOR that shown on the
figure.');
%         figure(7); imshow(BWLabelTTSelected);title('Selected Area');
%         pause(2);
        else
            disp('Breast is normal, there is no TUMOR.');
        end
end

arealist=zeros;
```

```
index=0;
selectedvalue=0;

end
end
end
BrstSample = java_array('java.lang.String', 48);   % number of photos in the folder

BrstSample(1) = java.lang.String('TUMOR');
BrstSample(2) = java.lang.String('TUMOR');
BrstSample(3) = java.lang.String('TUMOR');
BrstSample(4) = java.lang.String('TUMOR');
BrstSample(5) = java.lang.String('TUMOR');
BrstSample(6) = java.lang.String('TUMOR');
BrstSample(7) = java.lang.String('TUMOR');
BrstSample(8) = java.lang.String('TUMOR');
BrstSample(9) = java.lang.String('TUMOR');
BrstSample(10) = java.lang.String('TUMOR');
BrstSample(11) = java.lang.String('TUMOR');
BrstSample(12) = java.lang.String('TUMOR');
BrstSample(13) = java.lang.String('TUMOR');
BrstSample(14) = java.lang.String('TUMOR');
BrstSample(15) = java.lang.String('TUMOR');
BrstSample(16) = java.lang.String('TUMOR');
BrstSample(17) = java.lang.String('TUMOR');
BrstSample(18) = java.lang.String('TUMOR');
BrstSample(19) = java.lang.String('TUMOR');
BrstSample(20) = java.lang.String('TUMOR');
BrstSample(21) = java.lang.String('TUMOR');
BrstSample(22) = java.lang.String('TUMOR');
BrstSample(23) = java.lang.String('TUMOR');
BrstSample(24) = java.lang.String('TUMOR');
BrstSample(25) = java.lang.String('TUMOR');
BrstSample(26) = java.lang.String('TUMOR');
BrstSample(27) = java.lang.String('TUMOR');
BrstSample(28) = java.lang.String('NORMAL');
BrstSample(29) = java.lang.String('NORMAL');
BrstSample(30) = java.lang.String('NORMAL');
BrstSample(31) = java.lang.String('NORMAL');
BrstSample(32) = java.lang.String('NORMAL');
BrstSample(33) = java.lang.String('NORMAL');
BrstSample(34) = java.lang.String('NORMAL');
BrstSample(35) = java.lang.String('NORMAL');
BrstSample(36) = java.lang.String('NORMAL');
BrstSample(37) = java.lang.String('NORMAL');
BrstSample(38) = java.lang.String('NORMAL');
BrstSample(39) = java.lang.String('NORMAL');
BrstSample(40) = java.lang.String('NORMAL');
BrstSample(41) = java.lang.String('NORMAL');
```

```
BrstSample(42) = java.lang.String('NORMAL');
BrstSample(43) = java.lang.String('NORMAL');
BrstSample(44) = java.lang.String('NORMAL');
BrstSample(45) = java.lang.String('NORMAL');
BrstSample(46) = java.lang.String('NORMAL');
BrstSample(47) = java.lang.String('NORMAL');
BrstSample(48) = java.lang.String('NORMAL');
% ...

BrstSampleList = cell(BrstSample);

FeatureListTT =
horzcat(SImgAreaTT,SImgFilledAreaTT,SImgCentroidXTT,SImgCentroidYTT,SImgMajorAxisLengthTT,SImgMinorAxisLengthTT,SImgEccentricityTT,SImgOrientationTT,SImgSolidityTT,SImgEquivDiameterTT);

traindata=FeatureListTT(1:end,1:10);%%% 1. ve 2. feature'lara göre değerlendirildi. Diğerleri eklenebilir.
group=BrstSampleList(1:end);
figure(8);
svmStruct1 =
svmtrain(traindata,group,'Kernel_Function','quadratic','method','QP','showplot',true);
```

Appendix 2 - Matlab Code of Developed Software; Test Process

```
%% Breast Cancer Classification System -- Test Phase
%% Selection of Image For Test Process
cd('c:\Program Files\MATLAB\R2011b\work\Testdata\');
imagelistTest = dir('*.pgm');
numTest = numel(imagelistTest);
imdataTest = cell(1,numel(imagelistTest));

%%%% Making List of Image in the Train Folder
for kT=1:numTest imdataTest{kT} = imread(imagelistTest(kT).name); end
%%%% Declaration of Feature Space Variables for Segmented Area %%%%%%%
SImgAreaTTest(numTest,1)=zeros;
SImgCentroidXTTest(numTest,1)=zeros;
SImgCentroidYTTest(numTest,1)=zeros;
SImgMajorAxisLengthTTest(numTest,1)=zeros;
SImgMinorAxisLengthTTest(numTest,1)=zeros;
SImgEccentricityTTest(numTest,1)=zeros;
SImgOrientationTTest(numTest,1)=zeros;
SImgFilledAreaTTest(numTest,1)=zeros;
SImgSolidityTTest(numTest,1)=zeros;
SImgEquivDiameterTTest(numTest,1)=zeros;

 BrstSampleList1 = cell(numTest,1);

%% Image Enhacement Process
%% Orijinal Image

for mT=1:numTest,
disp(mT);
ImportedImgTest = imdataTest{mT};

ImgTTest = ImportedImgTest;
%figure(1);imshow(ImgTTest,'InitialMagnification',50);title('Original Image');
%pause(0.5);
%close();

%% Converted to Binary (Black&White) Image of Orijinal Image
ImgTTestBWU = im2bw(ImgTTest,0.70);
%figure(2);imshow(ImgTTestBWU,'InitialMagnification',50);title('Binary Version of
Original Image');
%pause(0.5);
%close();

%% Converted to Binary (Black&White) Image and Selected Big Areas Image %%%
ImgBWUUSegTest = bwareaopen(ImgTTestBWU, 10000);
```

```
%figure(3);imshow(ImgBWUUSegTest,'InitialMagnification',50);title('Binary Version
Areas of Original Image');
%pause(0.5);
%close();
%% Binary (Black&White) Image Selected Area is Filled
ImgBWSegUUFilTest= imfill(ImgBWUUSegTest,'holes');
%figure(4);imshow(ImgBWSegUUFilTest,'InitialMagnification',50);title('Filled up Areas
of B/W Image');
%pause(0.5);
%close();

%% Binary Image Selected Area Region Properties, BoundingBox Properties %%%
ImgStatsTrTest = regionprops(ImgBWSegUUFilTest, 'BoundingBox'); %BoundingBox
Prop
ImgStatSizeTrTest=size(ImgStatsTrTest);                %List of Output Stats
ImgStatSizeTrXTest=ImgStatSizeTrTest(1);               %No of Picked Areas

for ii=1:ImgStatSizeTrXTest                        %Chosing right
        BoxListTest{ii}=ImgStatsTrTest(ii).BoundingBox;     %BoundingBox
        if((ImgStatsTrTest(ii).BoundingBox(2))<10)      %BoundingBox 2nd value
           selectedBoxTrIdxTest=ii;                %If it is <10, OK.
           selectedBoxTrTest=BoxListTest{ii};
        end
end

%% Crop BoundingBoxed Image From Original Image
ImgUUCropTest=
imcrop(ImgTTest,ImgStatsTrTest(selectedBoxTrIdxTest).BoundingBox);
%figure(5);imshow(ImgUUCropTest,'InitialMagnification',50);title('Cropped Region of
Original Image');
%pause(0.5);
%close();
%% Declaration of Variables for Removing BoundingBoxed Region/Area
xTest=real(zeros);
yTest=real(zeros);
xlTest=real(zeros);
ylTest=real(zeros);
xpTest=real(zeros);
ypTest=real(zeros);
%%%%%%%%%%%%%
xTest=(round(ImgStatsTrTest(ii).BoundingBox(1)))-30;
yTest=(round(ImgStatsTrTest(ii).BoundingBox(2)));
xlTest=xTest+(round(ImgStatsTrTest(ii).BoundingBox(3)))+30;
ylTest=yTest+(round(ImgStatsTrTest(ii).BoundingBox(4)))+150;
xpTest=xTest+1; ypTest=yTest+10;
xloopTest=xlTest-xTest;
yloopTest=(ylTest-yTest)/(ypTest-yTest);
%%%%%%%%%%%%%
% Removes BoundingBoxed Region/Area On the Original Image Pixels -> Black %
```

```
if((round(ImgStatsTrTest(ii).BoundingBox(1)))>300)
          for jjTest=1:yloopTest
          ImgTTest(yTest:ypTest,xpTest:xlTest)=0;
          xpTest=xpTest+4;
                yTest=yTest+10;
                ypTest=ypTest+10;
   end
else
          for jTest=1:yloopTest
       ImgTTest(yTest:ypTest,xpTest:xlTest)=0;
       xlTest=xlTest-4;
             yTest=yTest+10;
             ypTest=ypTest+10;
          end
end
%%%%%%%%%%%%%%
xTest=real(zeros);
yTest=real(zeros);
xlTest=real(zeros);
ylTest=real(zeros);
xpTest=real(zeros);
ypTest=real(zeros);
xloopTest=real(zeros);
yloopTest=real(zeros);
%%%%%%%%%%%%%%
% Applying Gaussian Smoothing Filter on the modified Image
%%%%%%%%%%%%%%%%
HTTrTest = fspecial('gaussian',[5 5],1);
ImgGFTTest = imfilter(ImgTTest,HTTrTest,'replicate','same');
% figure(6); imshow(ImgGFTTest,'InitialMagnification',50);title('Gaussian Filtered
Image');
% pause(0.5);
% close();
%%%%%%%%%%%%%%
% Applying Contrast Streching on the modified Image
%%%%%%%%%%%%%%%%%%%%%%%
ImgCSTTest = imadjust(ImgGFTTest,[0.45 1],[]); %stretchlim(ImgGF)
% figure(7), imshow(ImgCSTTest,'InitialMagnification',50), title('Contrast Streching of
Image');
% pause(0.5);
% close();
%%%%%%%%%%%
% Applying Top Hat Filtering on the modified Image
%%%%%%%%%%%%%%%%%%%%%%%%
seTTest = strel('disk',250);
ImgTHFTTest = imtophat(ImgCSTTest,seTTest);
% figure(8), imshow(ImgTHFTTest,'InitialMagnification',50), title('Tophat filtering of
Image');
% pause(0.5);
```

```
% close();
%%%%%%%%%%%
% Applying Wavelet Filtering on the modified Image
%%%%%%%%%%%%%%%%%%%%%%%%%
% Two Scale 2D Discrete Wavelet Transform Daubechies, Decomp&Reconst %%%%%%
% Load Filtered Image for Wavelet Transform Decomp&Reconst
%%%%%%%%%%%%%%%%%
Xtest=ImgTHFTTest;
%% Set the level for decomposition.Compute 2-level decomposition of img %%%
NTest=2;
[CTest,STest] = wavedec2(Xtest,NTest,'db1');
%%%% Extract the level 1 coefficients
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cA1Test = appcoef2(CTest,STest,'db1',1);
[cH1Test,cV1Test,cD1Test] = detcoef2('all',CTest,STest,1);
%%%% Extract the level 2 coefficients
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cA2Test = appcoef2(CTest,STest,'db1',2);
[cH2Test,cV2Test,cD2Test] = detcoef2('all',CTest,STest,2);
%%%% Here are reconstructed branches
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A1Test = wrcoef2('a',CTest,STest,'db1',1);
A2Test = wrcoef2('a',CTest,STest,'db1',2);
H1Test = wrcoef2('h',CTest,STest,'db1',1);
V1Test = wrcoef2('v',CTest,STest,'db1',1);
D1Test = wrcoef2('d',CTest,STest,'db1',1);
H2Test = wrcoef2('h',CTest,STest,'db1',2);
V2Test = wrcoef2('v',CTest,STest,'db1',2);
D2Test = wrcoef2('d',CTest,STest,'db1',2);
%%%% Set the threshold
[thr,sorh,keepapp] = ddencmp('cmp','wv',Xtest);
[XC,CXC,LXC,PERF0,PERFL2]=wdencmp('gbl',CTest,STest,'db1',NTest,thr,sorh,keepap
p);
%%%% Multilevel 2-D wavelet reconstruction
M=waverec2(CXC,LXC,'db1');
out1trTest=uint8(M);
%%%% calculate the peak signal to noise ratio
input_ima1=double(Xtest);
    out2=double(out1trTest);
    error=0;
     for yTest=1:191
      for xTest=1:159
        MSE=((input_ima1(xTest,yTest))-(out2(xTest,yTest)))^2;
          error=MSE+error;
      end
     end
MSE_WO=(1/(159*191))*error;
PSNR_WO=20*log10(255/sqrt(MSE_WO));
%%%%subtracting the original and reconstructed image
```

```
diff_ima=imsubtract(input_ima1,out2);
%figure(9); imshow(out1trTest,'InitialMagnification',50); title('Image After Discrete
Wavelet Transform');
%pause(0.5);
%close();
% Segmentation Process after Filtering Image
ImgBWTTrTest=im2bw(out1trTest);
HSTTrTest = fspecial('gaussian',[5 5],1);
ImgGFBWTTrTest = imfilter(ImgBWTTrTest,HSTTrTest,'replicate','same');
% figure(10); imshow(ImgGFBWTTrTest,'InitialMagnification',50);title('Gaussian Filtered
and Segmented Image');
% pause(1);
% close();
% Classification Process after Filtering Image
BWLabelTTrTest = bwlabel(ImgGFBWTTrTest);
statsTest = regionprops(BWLabelTTrTest,'all');

ImgTrStatSizeTest=size(statsTest);
ImgTrStatSizeTestX=ImgTrStatSizeTest(1);

for iii=1:ImgTrStatSizeTestX
   ImgStatTrAreaListTest{iii}=statsTest(iii).Area;
end

if(ImgTrStatSizeTestX==0)

       SImgAreaTTest(kT,1)=zeros; %Segmented Image -Area Property
       SImgCentroidXTTest(kT,1)=zeros;
       SImgCentroidYTTest(kT,1)=zeros;
       SImgMajorAxisLengthTTest(kT,1)=zeros;
       SImgMinorAxisLengthTTest(kT,1)=zeros;
       SImgEccentricityTTest(kT,1)=zeros;
       SImgOrientationTTest(kT,1)=zeros;
       SImgFilledAreaTTest(kT,1)=zeros;
       SImgSolidityTTest(kT,1)=zeros;
       SImgEquivDiameterTTest(kT,1)=zeros;

%        disp('Breast is normal, there is no TUMOR.');

else

ImgStatAreaLstMatTrTest=cell2mat(ImgStatTrAreaListTest);
StatAreaMaxValueTrTest=max(ImgStatAreaLstMatTrTest);

if(StatAreaMaxValueTrTest<300)

       SImgAreaTTest(kT,1)=zeros; %Segmented Image -Area Property
       SImgCentroidXTTest(kT,1)=zeros;
       SImgCentroidYTTest(kT,1)=zeros;
```

```
        SImgMajorAxisLengthTTest(kT,1)=zeros;
        SImgMinorAxisLengthTTest(kT,1)=zeros;
        SImgEccentricityTTest(kT,1)=zeros;
        SImgOrientationTTest(kT,1)=zeros;
        SImgFilledAreaTTest(kT,1)=zeros;
        SImgSolidityTTest(kT,1)=zeros;
        SImgEquivDiameterTTest(kT,1)=zeros;

%        disp('Breast is normal, there is no TUMOR.');
else

if isempty(statsTest)

        SImgAreaTTest(kT,1)=zeros; %Segmented Image -Area Property
        SImgCentroidXTTest(kT,1)=zeros;
        SImgCentroidYTTest(kT,1)=zeros;
        SImgMajorAxisLengthTTest(kT,1)=zeros;
        SImgMinorAxisLengthTTest(kT,1)=zeros;
        SImgEccentricityTTest(kT,1)=zeros;
        SImgOrientationTTest(kT,1)=zeros;
        SImgFilledAreaTTest(kT,1)=zeros;
        SImgSolidityTTest(kT,1)=zeros;
        SImgEquivDiameterTTest(kT,1)=zeros;

%       disp('Breast is normal, there is no TUMOR.');

else
        [obj1cntTest,obj2cntTest] = size(statsTest);
        for stscntTest = 1:(obj1cntTest)
            arealistTest(stscntTest)=statsTest(stscntTest).Area;
        end
        MaxAreaTest = max(arealistTest);

range=300:7500;
selectedvalueTest=max(arealistTest(arealistTest>=min(range) &
arealistTest<=max(range)));

if isempty(selectedvalueTest)
    range=7501:800000;
    selectedvalueTest=max(arealistTest(arealistTest>=min(range) &
arealistTest<=max(range)));
end

if isempty(selectedvalueTest)

    range=0:300;
    selectedvalueTest=max(arealistTest(arealistTest>=min(range) &
arealistTest<=max(range)));
```

```
        SImgAreaTTest(kT,1)=zeros; %Segmented Image -Area Property
        SImgCentroidXTTest(kT,1)=zeros;
        SImgCentroidYTTest(kT,1)=zeros;
        SImgMajorAxisLengthTTest(kT,1)=zeros;
        SImgMinorAxisLengthTTest(kT,1)=zeros;
        SImgEccentricityTTest(kT,1)=zeros;
        SImgOrientationTTest(kT,1)=zeros;
        SImgFilledAreaTTest(kT,1)=zeros;
        SImgSolidityTTest(kT,1)=zeros;
        SImgEquivDiameterTTest(kT,1)=zeros;

%       disp('Breast is normal, there is no TUMOR.');

end

    indexTest=find(arealistTest==selectedvalueTest);
    indexTest=indexTest(1);

    Area=statsTest(indexTest).Area;
    SImgAreaTTest(mT,1)=Area;

    CentroidX=statsTest(indexTest).Centroid(1);
    SImgCentroidXTTest(mT,1)=CentroidX;

    CentroidY=statsTest(indexTest).Centroid(2);
    SImgCentroidYTTest(mT,1)=CentroidY;

    MajorAxisLength=statsTest(indexTest).MajorAxisLength;
    SImgMajorAxisLengthTTest(mT,1)=MajorAxisLength;

    MinorAxisLength=statsTest(indexTest).MinorAxisLength;
    SImgMinorAxisLengthTTest(mT,1)=MinorAxisLength;

    Eccentricity=statsTest(indexTest).Eccentricity;
    SImgEccentricityTTest(mT,1)=Eccentricity;

    Orientation=statsTest(indexTest).Orientation;
    SImgOrientationTTest(mT,1)=Orientation;

    FilledArea=statsTest(indexTest).FilledArea;
    SImgFilledAreaTTest(mT,1)=FilledArea;

    Solidity=statsTest(indexTest).Solidity;
    SImgSolidityTTest(mT,1)=Solidity;

    EquivDiameter=statsTest(indexTest).EquivDiameter;
    SImgEquivDiameterTTest(mT,1)=EquivDiameter;

        if (Area>=300 && Area <=7500)
```

```
        BWLabelTTSelectedTest = ismember(BWLabelTTrTest, indexTest);
%                             disp('Breast is not normal, there is a TUMOR that shown on the
figure.');
%         figure(7); imshow(BWLabelTTSelectedTest);title('Selected Area');
%         pause(2);
        else
%             disp('Breast is normal, there is no TUMOR.');
        end
end

arealistTest=zeros;
indexTest=0;
selectedvalueTest=0;
end
end
end

FeatureListTTest =
horzcat(SImgAreaTTest,SImgFilledAreaTTest,SImgCentroidXTTest,SImgCentroidYTTes
t,SImgMajorAxisLengthTTest,SImgMinorAxisLengthTTest,SImgEccentricityTTest,SImg
OrientationTTest,SImgSolidityTTest,SImgEquivDiameterTTest);

% % % % testData=[FeatureListTTest(1,1),FeatureListTTest(1,2)];
%testData=[FeatureListTTest(1,1),FeatureListTTest(1,2),FeatureListTTest(1,3),FeatureLis
tTTest(1,4),FeatureListTTest(1,5),FeatureListTTest(1,6),FeatureListTTest(1,7),FeatureList
TTest(1,8),FeatureListTTest(1,9),FeatureListTTest(1,10)];
% % % % %testData=FeatureListTTest;
% % % %
% % % % BrstSampleList = svmclassify(svmStruct1,testData,'showplot',true)
% % % % hold on;
% % % % plot(testData(1),testData(2),'ro','MarkerSize',12);
% % % % hold off
% % % % %hold
on;plot(testData(1),testData(2),testData(3),testData(4),testData(5),testData(6),testData(7),t
estData(8),testData(9),testData(10),'ro','MarkerSize',12);hold off

for kTT=1: numTest,
    disp(kTT);

testData=[FeatureListTTest(kTT,1),FeatureListTTest(kTT,2),FeatureListTTest(kTT,3),Fea
tureListTTest(kTT,4),FeatureListTTest(kTT,5),FeatureListTTest(kTT,6),FeatureListTTest(
kTT,7),FeatureListTTest(kTT,8),FeatureListTTest(kTT,9),FeatureListTTest(kTT,10)];
    BrstSampleList = svmclassify(svmStruct1,testData,'showplot',true)
%   hold on;
%       plot(testData(1),testData(2),'ro','MarkerSize',12);
%   hold off
    BrstSampleList1(kTT)= BrstSampleList(1,1);

end
```

Appendix 3 - Matlab Code of Developed Software; Classification Process

```
%% Breast Cancer Classification System
%% Selection of Image For Classification Process

[ImgNameC ImgPathC] = uigetfile ('C:\Program Files\MATLAB\R2011b\work\*.*' ,
'Load Image to Classify');
ImgFullPathC=strcat(ImgPathC, filesep, ImgNameC);
ImportedImgTestC= imread(ImgFullPathC);

SImgAreaTTestC(1,1)=zeros;
SImgCentroidXTTestC(1,1)=zeros;
SImgCentroidYTTestC(1,1)=zeros;
SImgMajorAxisLengthTTestC(1,1)=zeros;
SImgMinorAxisLengthTTestC(1,1)=zeros;
SImgEccentricityTTestC(1,1)=zeros;
SImgOrientationTTestC(1,1)=zeros;
SImgFilledAreaTTestC(1,1)=zeros;
SImgSolidityTTestC(1,1)=zeros;
SImgEquivDiameterTTestC(1,1)=zeros;
%% Image Enhacement Process
%% Orijinal Image
ImgTTestC = ImportedImgTestC;
%figure(1);imshow(ImgTTest,'InitialMagnification',50);title('Original Image');
%pause(0.5);
%close();
%% Converted to Binary (Black&White) Image of Orijinal Image
ImgTTestBWUC = im2bw(ImgTTestC,0.70);
%figure(2);imshow(ImgBW,'InitialMagnification',50);title('Binary Version of Original
Image');
%pause(0.5);
%close();
%% Converted to Binary (Black&White) Image and Selected Big Areas Image %%%
ImgBWUSegC = bwareaopen(ImgTTestBWUC, 10000);
%figure(3);imshow(ImgBWSeg,'InitialMagnification',50);title('Binary Version Areas of
Original Image');
%pause(0.5);
%close();
%% Binary (Black&White) Image Selected Area is Filled
ImgBWSegUFilC= imfill(ImgBWUSegC,'holes');
%figure(4);imshow(ImgBWSegFil,'InitialMagnification',50);title('Filled up Areas of B/W
Image');
%pause(0.5);
%close();
%%%%%%%
%% Binary Image Selected Area Region Properties, BoundingBox Properties %%%
ImgStatsC = regionprops(ImgBWSegUFilC, 'BoundingBox'); %BoundingBox Prop
ImgStatSizeC=size(ImgStatsC);                        %List of Output Stats
```

```
ImgStatSizeCX=ImgStatSizeC(1);                    %No of Picked Areas

for iiC=1:ImgStatSizeCX                           %Chosing right
      BoxList{iiC}=ImgStatsC(iiC).BoundingBox;    %BoundingBox
      if((ImgStatsC(iiC).BoundingBox(2))<10)      %BoundingBox 2nd value
        selectedBoxCIdx=iiC;                      %If it is <10, OK.
        selectedBoxC=BoxList{iiC};
      end
end
%%%%%%%%%
%% Crop BoundingBoxed Image From Original Image
ImgUCropC= imcrop(ImgTTestC,ImgStatsC(selectedBoxCIdx).BoundingBox);
%figure(5);imshow(ImgUCrop,'InitialMagnification',50);title('Cropped Region of Original
Image');
%pause(0.5);
%close();
%%%%%%%%%
%% Declaration of Variables for Removing BoundingBoxed Region/Area
%%%%%%%%
xC=real(zeros);
yC=real(zeros);
xlC=real(zeros);
ylC=real(zeros);
xpC=real(zeros);
ypC=real(zeros);
%%%%%%%%%
xC=(round(ImgStatsC(iiC).BoundingBox(1)))-30;
yC=(round(ImgStatsC(iiC).BoundingBox(2)));
xlC=xC+(round(ImgStatsC(iiC).BoundingBox(3)))+30;
ylC=yC+(round(ImgStatsC(iiC).BoundingBox(4)))+150;
xpC=xC+1; ypC=yC+10;
xloopC=xlC-xC;
yloopC=(ylC-yC)/(ypC-yC);
%%%%%%%%%
% Removes BoundingBoxed Region/Area On the Original Image Pixels -> Black %
if((round(ImgStatsC(iiC).BoundingBox(1)))>300)
       for jC=1:yloopC
       ImgTTestC(yC:ypC,xpC:xlC)=0;
       xpC=xpC+4;
         yC=yC+10;
         ypC=ypC+10;
     end
else
       for jC=1:yloopC
     ImgTTestC(yC:ypC,xpC:xlC)=0;
     xlC=xlC-4;
       yC=yC+10;
       ypC=ypC+10;
       end
```

```
end
%%%%%%%%%
% Applying Gaussian Smoothing Filter on the modified Image
%%%%%%%%%%%%%%%%
HTTestC = fspecial('gaussian',[5 5],1);
ImgGFTTestC = imfilter(ImgTTestC,HTTestC,'replicate','same');
% figure(6); imshow(ImgGFTTest,'InitialMagnification',50);title('Gaussian Filtered
Image');
% pause(0.5);
% close();
%%%%%%%%%
% Applying Contrast Streching on the modified Image
%%%%%%%%%%%%%%%%%%%%%%%
ImgCSTTestC = imadjust(ImgGFTTestC,[0.45 1],[]); %stretchlim(ImgGF)
% figure(7), imshow(ImgCSTTest,'InitialMagnification',50), title('Contrast Streching of
Image');
% pause(0.5);
% close();
%%%%%%%%%%
% Applying Top Hat Filtering on the modified Image
%%%%%%%%%%%%%%%%%%%%%%%%
seTTestC = strel('disk',250);
ImgTHFTTestC = imtophat(ImgCSTTestC,seTTestC);
% figure(8), imshow(ImgTHFTTest,'InitialMagnification',50), title('Tophat filtering of
Image');
% pause(0.5);
% close();
%%%%%%%%%%%
% Applying Wavelet Filtering on the modified Image
%%%%%%%%%%%%%%%%%%%%%%%%
% Two Scale 2D Discrete Wavelet Transform Daubechies, Decomp&Reconst %%%%%%
% Load Filtered Image for Wavelet Transform Decomp&Reconst
%%%%%%%%%%%%%%%%
XtestC=ImgTHFTTestC;
%% Set the level for decomposition.Compute 2-level decomposition of img %%%
N=2;
[CC,SC] = wavedec2(XtestC,N,'db1');
%%%% Extract the level 1 coefficients
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cA1 = appcoef2(CC,SC,'db1',1);
[cH1,cV1,cD1] = detcoef2('all',CC,SC,1);
%%%% Extract the level 2 coefficients
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cA2 = appcoef2(CC,SC,'db1',2);
[cH2,cV2,cD2] = detcoef2('all',CC,SC,2);
%%%% Here are reconstructed branches
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A1 = wrcoef2('a',CC,SC,'db1',1);
A2 = wrcoef2('a',CC,SC,'db1',2);
```

```
H1 = wrcoef2('h',CC,SC,'db1',1);
V1 = wrcoef2('v',CC,SC,'db1',1);
D1 = wrcoef2('d',CC,SC,'db1',1);
H2 = wrcoef2('h',CC,SC,'db1',2);
V2 = wrcoef2('v',CC,SC,'db1',2);
D2 = wrcoef2('d',CC,SC,'db1',2);
%%%% Set the threshold
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%
[thr,sorh,keepapp] = ddencmp('cmp','wv',XtestC);
[XC,CXC,LXC,PERF0,PERFL2]=wdencmp('gbl',CC,SC,'db1',N,thr,sorh,keepapp);
%%%% Multilevel 2-D wavelet reconstruction
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M=waverec2(CXC,LXC,'db1');
out1testC=uint8(M);
%%%% calculate the peak signal to noise ratio
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
input_ima1=double(XtestC);
     out2=double(out1testC);
     error=0;
      for yC=1:191
       for xC=1:159
         MSE=((input_ima1(xC,yC))-(out2(xC,yC)))^2;
           error=MSE+error;
       end
      end
MSE_WO=(1/(159*191))*error;
PSNR_WO=20*log10(255/sqrt(MSE_WO));
%%%%subtracting the original and reconstructed image
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
diff_ima=imsubtract(input_ima1,out2);

%figure(9); imshow(out1testC,'InitialMagnification',50); title('Image After Discrete
Wavelet Transform');
%pause(0.5);
%close();

% Segmentation Process after Filtering Image
ImgBWTTestC=im2bw(out1testC);
HSTTesttC = fspecial('gaussian',[5 5],1);
ImgGFBWTTestC = imfilter(ImgBWTTestC,HSTTesttC,'replicate','same');
% figure(10); imshow(ImgGFBWTTest,'InitialMagnification',50);title('Gaussian Filtered
and Segmented Image');
% pause(1);
% close();
% Classification Process after Filtering Image
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
BWLabelTTestC = bwlabel(ImgGFBWTTestC);
statsC = regionprops(BWLabelTTestC,'all');
```

```
ImgTestStatSizeC=size(statsC);
ImgTestStatSizeCX=ImgTestStatSizeC(1);

for iiiC=1:ImgTestStatSizeCX
   ImgStatAreaListC{iiiC}=statsC(iiiC).Area;
end

if(ImgTestStatSizeCX==0)

        SImgAreaTTestC(1,1)=zeros; %Segmented Image -Area Property
        SImgCentroidXTTestC(1,1)=zeros;
        SImgCentroidYTTestC(1,1)=zeros;
        SImgMajorAxisLengthTTestC(1,1)=zeros;
        SImgMinorAxisLengthTTestC(1,1)=zeros;
        SImgEccentricityTTestC(1,1)=zeros;
        SImgOrientationTTestC(1,1)=zeros;
        SImgFilledAreaTTestC(1,1)=zeros;
        SImgSolidityTTestC(1,1)=zeros;
        SImgEquivDiameterTTestC(1,1)=zeros;

%         disp('Breast is normal, there is no TUMOR.');

else

ImgStatAreaLstMatC=cell2mat(ImgStatAreaListC);
StatAreaMaxValueC=max(ImgStatAreaLstMatC);

if(StatAreaMaxValueC<300)

        SImgAreaTTestC(1,1)=zeros; %Segmented Image -Area Property
        SImgCentroidXTTestC(1,1)=zeros;
        SImgCentroidYTTestC(1,1)=zeros;
        SImgMajorAxisLengthTTestC(1,1)=zeros;
        SImgMinorAxisLengthTTestC(1,1)=zeros;
        SImgEccentricityTTestC(1,1)=zeros;
        SImgOrientationTTestC(1,1)=zeros;
        SImgFilledAreaTTestC(1,1)=zeros;
        SImgSolidityTTestC(1,1)=zeros;
        SImgEquivDiameterTTestC(1,1)=zeros;

%         disp('Breast is normal, there is no TUMOR.');
else

if isempty(statsC)

        SImgAreaTTestC(1,1)=zeros; %Segmented Image -Area Property
        SImgCentroidXTTestC(1,1)=zeros;
        SImgCentroidYTTestC(1,1)=zeros;
```

```
        SImgMajorAxisLengthTTestC(1,1)=zeros;
        SImgMinorAxisLengthTTestC(1,1)=zeros;
        SImgEccentricityTTestC(1,1)=zeros;
        SImgOrientationTTestC(1,1)=zeros;
        SImgFilledAreaTTestC(1,1)=zeros;
        SImgSolidityTTestC(1,1)=zeros;
        SImgEquivDiameterTTestC(1,1)=zeros;

%         disp('Breast is normal, there is no TUMOR.');

else
        [obj1cntC,obj2cntC] = size(statsC);
        for stscntC = 1:(obj1cntC)
            arealistC(stscntC)=statsC(stscntC).Area;
        end
        MaxArea = max(arealistC);

range=300:7500;
selectedvalueC=max(arealistC(arealistC>=min(range) & arealistC<=max(range)));

if isempty(selectedvalueC)
    range=7501:800000;
    selectedvalueC=max(arealistC(arealistC>=min(range) & arealistC<=max(range)));
end

if isempty(selectedvalueC)

    range=0:300;
    selectedvalueC=max(arealistC(arealistC>=min(range) & arealistC<=max(range)));

        SImgAreaTTestC(k,1)=zeros; %Segmented Image -Area Property
        SImgCentroidXTTestC(k,1)=zeros;
        SImgCentroidYTTestC(k,1)=zeros;
        SImgMajorAxisLengthTTestC(k,1)=zeros;
        SImgMinorAxisLengthTTestC(k,1)=zeros;
        SImgEccentricityTTestC(k,1)=zeros;
        SImgOrientationTTestC(k,1)=zeros;
        SImgFilledAreaTTestC(k,1)=zeros;
        SImgSolidityTTestC(k,1)=zeros;
        SImgEquivDiameterTTestC(k,1)=zeros;


%         disp('Breast is normal, there is no TUMOR.');

end

     indexC=find(arealistC==selectedvalueC);
     indexC=indexC(1);
```

```
    Area=statsC(indexC).Area;
    SImgAreaTTestC(1,1)=Area;

    CentroidX=statsC(indexC).Centroid(1);
    SImgCentroidXTTestC(1,1)=CentroidX;

    CentroidY=statsC(indexC).Centroid(2);
    SImgCentroidYTTestC(1,1)=CentroidY;

    MajorAxisLength=statsC(indexC).MajorAxisLength;
    SImgMajorAxisLengthTTestC(1,1)=MajorAxisLength;

    MinorAxisLength=statsC(indexC).MinorAxisLength;
    SImgMinorAxisLengthTTestC(1,1)=MinorAxisLength;

    Eccentricity=statsC(indexC).Eccentricity;
    SImgEccentricityTTestC(1,1)=Eccentricity;

    Orientation=statsC(indexC).Orientation;
    SImgOrientationTTestC(1,1)=Orientation;

    FilledArea=statsC(indexC).FilledArea;
    SImgFilledAreaTTestC(1,1)=FilledArea;

    Solidity=statsC(indexC).Solidity;
    SImgSolidityTTestC(1,1)=Solidity;

    EquivDiameter=statsC(indexC).EquivDiameter;
    SImgEquivDiameterTTestC(1,1)=EquivDiameter;

        if (Area>=300 && Area <=7500)
        BWLabelTTestSelectedC = ismember(BWLabelTTestC, indexC);
%                           disp('Breast is not normal, there is a TUMOR that shown on the
figure.');
        figure(11); imshow(BWLabelTTestSelectedC,'InitialMagnification',50);title('Selected
Area of Image');
        pause(2);
        else
%           disp('Breast is normal, there is no TUMOR.');
        end
end
end
end
FeatureListTTestC =
horzcat(SImgAreaTTestC,SImgFilledAreaTTestC,SImgCentroidXTTestC,SImgCentroidY
TTestC,SImgMajorAxisLengthTTestC,SImgMinorAxisLengthTTestC,SImgEccentricityT
TestC,SImgOrientationTTestC,SImgSolidityTTestC,SImgEquivDiameterTTestC);

testDataC=[FeatureListTTest(1,1),FeatureListTTest(1,2)];
```

```
%testData=[FeatureListTTestC(1,1),FeatureListTTestC(1,2),FeatureListTTestC(1,3),Featu
reListTTestC(1,4),FeatureListTTestC(1,5),FeatureListTTestC(1,6),FeatureListTTestC(1,7)
,FeatureListTTestC(1,8),FeatureListTTestC(1,9),FeatureListTTestC(1,10)];
%testData=FeatureListTTest;

BrstSampleList = svmclassify(svmStruct1,testDataC,'showplot',true)
hold on;
plot(testDataC(1),testDataC(2),'ro','MarkerSize',12);
hold off
%hold
on;plot(testData(1),testData(2),testData(3),testData(4),testData(5),testData(6),testData(7),t
estData(8),testData(9),testData(10),'ro','MarkerSize',12);hold off
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %
%%%%%End of  Program%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```