ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

TISSUE DENSITY CLASSIFICATION IN MAMMOGRAPHIC IMAGES USING LOCAL FEATURES

M.Sc. THESIS

Sezer KUTLUK

Department of Electronics and Communications Engineering

Biomedical Engineering Programme

JUNE 2012

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

TISSUE DENSITY CLASSIFICATION IN MAMMOGRAPHIC IMAGES USING LOCAL FEATURES

M.Sc. THESIS

Sezer KUTLUK
(504091424)

Department of Electronics and Communications Engineering

Biomedical Engineering Programme

Thesis Advisor: Prof. Dr. Bilge GÜNSEL

JUNE 2012

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

YEREL ÖZNİTELİKLER İLE MAMOGRAFİ GÖRÜNTÜLERİNDE DOKU YOĞUNLUĞUNUN SINIFLANDIRILMASI

YÜKSEK LİSANS TEZİ

Sezer KUTLUK
(504091424)

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Biyomedikal Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Bilge GÜNSEL

HAZİRAN 2012

Sezer KUTLUK, an **M.Sc.** student of **ITU Graduate School of Science Engineering and Technology** with student ID **504091424**, successfully defended the thesis entitled "**TISSUE DENSITY CLASSIFICATION IN MAMMOGRAPHIC IMAGES USING LOCAL FEATURES**", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Bilge GÜNSEL** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. Aydın AKAN** Istanbul University	.....
	Asst. Prof. Dr. Mustafa Ersel KAMAŞAK Istanbul Technical University	.....

Date of Submission : 4 May 2012
Date of Defense : 5 June 2012

To my family,

FOREWORD

I would like to thank Prof. Dr. Bilge Günsel, Prof. Dr. Aydın Akan, Asst. Prof. Dr. Mustafa E. Kamaşak, Assoc. Prof. Dr. Neslihan S. Şengör, Özgün Çırakman, Ömer Deniz Akyıldız, Ozan Gürsoy, Serap Kırbız, Berat Denizdurduran, Onur Ural and my family for all the things they did.

June 2012

Sezer KUTLUK
(Electrical-Electronics Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

CAD	**:** Computer Aided Diagnosis
CBMIR	**:** Content Based Medical Image Retrieval
GMM	**:** Gaussian Mixture Models
kNN	**:** k-Nearest Neighbors
LVQ	**:** Learning Vector Quantization
SIFT	**:** Scale-Invariant Feature Transform
SVD	**:** Singular Value Decomposition
SVM	**:** Support Vector Machines

LIST OF TABLES

Page

LIST OF FIGURES

Page

TISSUE DENSITY CLASSIFICATION IN MAMMOGRAPHIC IMAGES USING LOCAL FEATURES

SUMMARY

In this study a breast tissue density classification system is proposed. Tissue density is known to be in high correlation to the development and diagnosis of breast cancer. The probability of a malignant mass occuring in some types of breast tissue is higher than others. Moreover, some tissue types hide the masses in mammographic images. Thus early detection of cancer, which has a key role in diagnosis, is obstructed.

Computer aided diagnosis (CAD) and content based medical image retrieval (CBMIR) systems may take advantage of breast tissue density classification since it can augment the performance, reliability and automaticity of these systems. Automatic abnormality detection systems get insensitive with increasing breast tissue density since dense tissue may hide tumors and microcalcifications. CAD systems may use tissue density information in determining the method for mass detection. CBMIR systems may use tissue density classification as a pre-elimination step, which decreases the processing time.

We use the MIAS dataset for our experiments, which is a widely used dataset in abnormality detection and density classification studies. Another reason for us to choose this dataset is that it provides a detailed groundtruth with annotations of density type and abnormality presence, type and location. This dataset contains mammographic images which are from three tissue density categories, namely fatty, fatty-glandular and dense-glandular. There are 322 images from 161 subjects in this dataset.

Mammographic images have a textural structure. Global image characteristics and the visual content in some images from different density classes are similar while some images from the same class have different characteristics. For this reason, we use local image features. We employ the scale-invariant feature transform for the extraction of local image features and apply a bag-of-features representation in order to model the data and select the training data optimally.

Classification of the extracted image features are performed using three different supervised classification methods, namely, Gaussian mixture models, support vector machines and learning vector quantization. By evaluating these three classifiers, we look for the optimal classification method for our problem. These methods are used to design classifiers by parametric, nonparametric and learning based approaches.

Several experiments were performed and different aspects of the system such as classification accuracy and dependence on the data size were evaluated. Feature based classification results in a 10-fold cross validation scheme as well as in a separate training and test sets scheme are reported. The effects of data size is observed and

reported by using an enlarged dataset. First experiments were performed in a two-class classification scheme in order to determine which classes are separable and which classes are hard to separate. Then three-class classification tests were performed and results are reported in a comparative manner.

Our results are promising that the developed system may be used as a building block of computer aided diagnosis and content based medical image retrieval systems.

YEREL ÖZNİTELİKLER İLE MAMOGRAFİ GÖRÜNTÜLERİNDE DOKU YOĞUNLUĞUNUN SINIFLANDIRILMASI

ÖZET

Bu çalışmada mamografi görüntülerinde göğüs dokusu yoğunluğunun sınıflandırılması amaçlı bir sistem önerilmiştir. Geliştirilen yöntemle görüntüler üç sınıfa ayrılmakta olup bu sınıflar yağlı doku, yağlı-bezel doku ve yoğun-bezel dokudur.

Doku yoğunluğunun göğüs kanseri oluşumunda ve kanser tanı sürecinde önemli bir parametre olduğu bilinmektedir. Yapılan çalışmalarda bazı doku yoğunluğu türlerinde kanser oluşma olasılığının diğerlerine göre daha yüksek olduğu belirtilmiştir. Bazı doku younluğu türlerinin de mamografi görüntülerinde tümör ve mikrokalsifikasyon oluşumlarının görünmesini engellediği, dolayısıyla erken kanser tanısına engel olarak tedavi sürecini geciktirdiği belirtilmektedir. Meme kanserinde erken tanının önemi düşünüldüğünde, doku yoğunluğunun sınıflandırılmasının büyük bir öneme sahip olduğu anlaşılmaktadır.

Mamografi görüntülerinde doku yoğunluğunun sınıflandırılması bilgisayarlı tanı sistemleri ve içerik tabanlı medikal görüntü arama sistemlerinin performans, doğruluk ve güvenilirliğini arttıracağı ve otomatikleştirilmesine katkıda bulunacağı için bu sistemlerde bir ön işlem bloğu olarak kullanılabilir. Otomatik kitle bulma uygulamalarının hassaslığı artan doku yoğunluğuyla tümör ve mikrokalsifikasyonlar dokunun içine gizlenebildiği için azalmaktadır. Doku yoğunluğu bilgisi kanser oluşumu ve tanı süreciyle ilgisinden dolayı bilgisayarlı tanılama sistemlerinde kullanılarak bu sistemlerin doğruluğu arttırılabilir. İçerik tabanlı medikal görüntü arama sistemlerinde ise, aranacak görüntü kümesini bir ön arama veya ek bir arama parametresi olarak azaltacak ve böylece hem arama doğruluğunu arttıracak, hem de arama getirme iş yükünü ve süresini önemli ölçüde azaltacaktır.

Bu çalışmada göğüs dokusunda kitle sezimi ve doku yoğunluğunun sınıflandırılması konulu çalışmalarda sıkça kullanılan MIAS görüntü veritabanı kullanılmıştır. Bu veritabanının kullanılmasındaki bir diğer neden de görüntülerle ilgili ayrıntılı bilginin veritabanını oluşturan grup tarafından sağlanmış olmasıdır. Bu bilgiler her görüntü için doku yoğunluğunu, eğer varsa doku içindeki kitlelerin türünü (iyicil ya da kötücül), büyüklüğünü ve doku içindeki konumunu içermektedir. Bu veritabanındaki görüntüler yağlı, yağlı-bezel ve yoğun-bezel olarak üçe ayrılmaktadır. Bir kişiden sağ ve sol olmak üzere iki mamografi görüntüsü alınmış olup bu veritabanında toplamda 322 görüntü vardır.

Mamografi görüntüleri dokusal yapıya sahiptir. Parlaklık, karşıtlık gibi genel görüntü özellikleri ve görsel içerik farklı sınıftan görüntülerde benzer olabilirken, aynı sınıftan görüntülerde de bu özelliklerin farklı olabildiği gözlenmiştir. Bir mamografi görüntüsü incelendiğinde dokudaki dağılımın düzgün olmadığı görülebilir. Bir sınıftaki

görüntülerin bazıları başka bir sınıfın görsel özelliklerine sahip olabilmektedir. Burada görsel özelliklerden kast edilen gri düzeyi histogramı ile parlak bölgelerin dağılımı ve yoğunluğudur.

Mamografi görüntülerindeki sınıf içi çeşitlilikten ve görüntülerin niteliğinden dolayı global öznitelik çıkarımı yöntemlerinin kullanılması durumunda iyi bir başarım elde edilemeyeceği düşünülmüş ve yerel öznitelik çıkarımı yöntemlerinin kullanılmasına karar verilmiştir. Global öznitelik çıkarımı yöntemleriyle doku içindeki dağılım bilgisi kullanılamayacak, farklı sınıflardaki benzer görsel özelliklere sahip görüntülerin ayırt edilmesi güçleşecektir.

Yerel görüntü öznitelik çıkarımı için önerilmiş birçok yöntem mevcuttur. Ölçekten Bağımsız Öznitelik Dönüşümü (SIFT), Hızlandırılmış Gürbüz Öznitelikler (SURF), Gradyan Histogramı (HOG) gibi birçok yöntem bu amaç için kullanılmaktadır. Bu yöntemler görüntünün tamamı yerine yerel inceleme yapıp önemli noktalar bulmaya çalışır. Bulunan her önemli nokta için bir öznitelik vektörü üretilir. Bu öznitelik vektörlerinin tümü veya seçilmiş bir bölümü görüntünün temsil edilmesi ve sınıflandırılması için kullanılabilir.

Bu çalışmada Ölçekten Bağımsız Öznitelik Dönüşümü (SIFT) metodu öznitelik çıkarımı amacıyla kullanılmıştır. SIFT algoritmasıyla bir görüntüden çok sayıda öznitelik çıkarılabilir. Çıkarılan öznitelik vektörlerinin her biri 128 boyutludur. Öznitelik çıkarımı için görüntü farklı ölçeklerde incelenerek önemli noktalar bulunur. Görüntünün farklı ölçeklerde incelenmesi için Gauss'ların Farkı yöntemi kullanılır. Bu yöntemde görüntüye farklı varyanslı Gauss filtreleri uygulanır. Bu filtrelerin uygulanmasıyla görüntü farklı miktarlarda bulanıklaştırılmış olur. Bu görüntülerin farkı alınarak görüntüdeki kenarlar ve köşeler elde edilir. Bu farkların bazı yöntemlerle elenmesiyle önemli noktalar bulunur. Önemli noktalarda ve komşularında gradyanlar hesaplanır. Her önemli nokta için bir genlik ve yön bilgisi hesaplanır. Bir Gauss penceresi kullanılarak önemli noktaya yakın olan noktaların etkisi arttırılırken, uzak olanlarınki azaltılır. Hesaplanan yön histogramları kullanılarak öznitelik vektörleri elde edilir.

Optimal öznitelik seçimi sınıflandırıcı tasarımında çok önemli bir adımdır. Özniteliklerin modellenmesi ve sınıflandırıcıların eğitileceği en iyi öznitelik kümesinin seçimi için öznitelik gruplama yöntemi kullanılmıştır. Bu gruplama öbekleme ile yapılmıştır. Yüksek bir öbek sayısıyla başlanmış ve yakın olan öbekler birleştirilerek optimum öbek sayısı elde edilmiştir. Öznitelik gruplama yönteminin kullanılmasındaki amaç eğitim kümesinin verideki tüm çeşitliliği yansıtabilmesini, her alt gruptan örnekler barındırmasını sağlamaktır. Böylece sınıflandırıcının verideki çeşitliliğin göz önüne alınarak tasarlanması ve test başarımının arttırılması sağlanmış olur.

Görüntülerden çıkarılan özniteliklerin sınıflandırılması için üç farklı eğiticili sınıflandırma metodu kullanılmıştır. Bu metodlar Gauss karışım modeli (GMM), destek vektör makinesi (SVM) ve öğrenmeli vektör seviyelemedir (LVQ). Üç farklı yöntem kullanılmasındaki amaç bu problem ve veri kümesi için en uygun sınıflandırıcının bulunmasıdır. Bu sınıflandırıcılar parametrik, parametrik olmayan ve öğrenme tabanlı yaklaşımlarla eğitim aşamasında eğitim kümesinden bir model oluşturur ve sınıflandırma aşamasında hangi sınıftan olduğu bilinmeyen yeni örnekleri sınıflandırır.

Gauss karışım modeli yönteminde her sınıf birden çok Gauss dağılımının birleşimiyle modellenmeye çalışılır. Her karışımın parametreleri, yani ortalama vektörleri ve kovaryans matrisleri, Beklenti-En Büyükleme (EM) algoritmasıyla kestirilir. Her sınıf için bir model oluşturulduktan sonra, yeni bir örneğin sınıflandırılması için bu karışımlar kullanılarak birer olasılık değeri hesaplanır. Gözlem en büyük olasılık değerinin elde edildiği sınıfa atanır.

Destek vektör makinesi yöntemi iki sınıftaki öznitelik vektörlerinin ortasındaki optimal hiperdüzlemi bulmaya çalışır. Bu yöntem temelde doğrusal olarak ayrıştırılabilen iki sınıfın sınıflandırılması için önerilmiştir; ancak bazı ek işlemlerle daha çok sınıf için, çekirdek fonksiyonlarının kullanımıyla da doğrusal olarak ayrıştırılamayan veri kümelerinde kullanılabilmektedir. Bunun için, doğrusal olarak sınıflandırılamayan veri bir çekirdek fonksiyonuyla doğrusal olarak sınıflandırılabileceği bir uzaya taşınır.

Öğrenmeli vektör seviyeleme yönteminde tasarım sınıflandırılmış özniteliklerin öbeklenmesiyle yapılır. Eğitim sırasında özniteliklerin etiketleri bilindiğinden, öbekleme iteratif olarak tekrarlanarak veri öğrenilir. Her öbek birden çok kodvektörü ile tanımlanır. Bu öğrenme bir ödül-ceza sistemine dayanır. Eğitim kümesindeki öznitelik vektörlerinin hangi sınıfa ait olduğu bilindiğinden, bir özniteliğin atandığı kodvektörü eğer doğru sınıftansa ödüllendirilir, yanlış sınıftansa cezalandırılır. Böylece, iteratif olarak en uygun öbek yapısına ulaşılmaya çalışılır. Bu yöntemde bir sınıfın temsil edileceği kodvektörü sayısının ve iterasyon sayısının belirlenmesi önemlidir.

Çeşitli deneylerle geliştirilen sistemin sınıflandırma doğruluğu ve eğitim kümesinin boyutuna bağlılık gibi özellikleri sınanmıştır. Dört farklı deney kurgulanmış ve bunlarla sınıflandırıcıların başarımı değerlendirilmiştir. 10 katlı çapraz geçerlilik testi ile ayrık eğitim-test kümelerinin kullanıldığı öznitelik tabanlı deneylerin sonuçları raporlanmıştır. 10 katlı çapraz geçerlilik testinin amacı, verinin her bölümünü eğitim ve test aşamalarında kullanmak, elde edilen sonuçların ortalamasının alınmasıyla verideki uç değerlerin etkisini azaltmaktır.

Eğitim kümesinin büyüklüğü daha büyük veri kümeleriyle yapılan deneylerle gözlenmiş ve raporlanmıştır. Bunun için eğitim kümesine yeni öznitelik vektörleri eklenmiş ve aynı öznitelik vektörleri tekrar tekrar eğitim için kullanılmıştır.

İki sınıflı sınıflandırma deneyleriyle sınıfların ayrıştırılabilme seviyeleri ile birbiriyle çokça karıştırılan ve iyi ayrılan sınıflar belirlenmiştir. Üç sınıflı sınıflandırma yapılarak da genel başarım raporlanmıştır.

Gauss karışım modeli sınıflandırıcısı kullanıldığında elde edilen sonuçlar kabul edilebilir sınırların altında kalmıştır. Bunun sebeplerinden biri, hesaplama karmaşıklığı arttığı için bir karışımın oluşturulduğu bileşen sayısı sınırlanmıştır. Bu da verinin iyi modellenememesine yol açmıştır.

Eğitim kümesindeki öznitelik vektörü sayısı az olduğunda destek vektör makinesi diğer yöntemlerden daha iyi başarım sağlamıştır.

Öğrenmeli vektör seviyeleme yöntemi eğitim kümesi küçük olduğunda düşük başarım gösterse de genişletilmiş eğitim kümesi kullanılarak tasarlandığında başarımı oldukça yükselmektedir. Eğitim kümesi genişletilirken uygulanan veri tekrarlama işlemi öğrenmeli vektör seviyeleme yönteminin başarımını olumlu yönde etkilemiştir. Daha

çok sayıda öznitelik vektörünün öğrenilmesi için kodvektörü sayısının ve iterasyon sayısının da arttırılması gerektiği gözlenmiştir.

Eğitim kümesini genişletmenin destek vektör makinesi yönteminde başarımı çok değiştirmediği, veri tekrarlamanın öğrenmeli vektör seviyeleme yöntemindeki kadar etkili olmadığı belirlenmiştir. Gauss karışım modeli ile test süreleri arttığından genişletilmiş eğitim kümesi kullanılmamıştır.

İki sınıflı sınıflandırıcılarla yapılan testlerden sonra, destek vektör makinesi ve öğrenmeli vektör seviyeleme yöntemleri kullanılarak üç sınıflı sınıflandırma testleri yapılmıştır. Bu testler yine genişletilmiş eğitim kümesi üzerinde, 10 katlı çapraz geçerlilik testi yöntemiyle yapılmıştır. Destek vektör makinesi kullanıldığında başarım düşerken, öğrenmeli vektör seviyeleme yöntemi kullanıldığında başarımın iki sınıflı durumda elde edilen sonuçlara yakın olduğu gözlenmiştir.

Deney sonuçları geliştirilen sistemin bilgisayarlı tanılama ve içerik tabanlı medikal görüntü arama getirme sistemlerinde kullanılabileceği konusunda umut vericidir.

1. INTRODUCTION

Research and development on computer aided diagnosis (CAD) systems aim to decrease the need for radiologists to read the medical images by providing fast and reliable information gathering from these images. Because an erroneous decision may directly affect a human's health, CAD systems are used only to assist the radiologists for the moment [1], while research on such systems is going on.

Tissue density in mammographic images is an important information for CAD systems. A simple scenario for a CAD system working on mammographic images is the automatic detection and classification of the cancerous tissue. Such a system extracts information from the given image and reports whether a tumor exists; and if it exists, the type, namely benign or malignant, the location and dimensions of the tumor are also reported. Classifying the tissue density has a big role in the accuracy of these information.

Additionally, tissue density classification provides the chance of selecting proper preprocessing steps. Image filters, contrast enhancement methods, segmentation methods and normalization processes can be applied up to the specific needs of the image in question.

A decrease in the sensitivity of automatic abnormality detection systems working on mammographic images with increasing breast tissue density are reported in several studies [2].

According to the American College of Radiology (ACR) Breast Imaging Reporting and Data System (BIRADS), breast density is classified into four categories [3]:

- BIRADS I: the breast is almost entirely fatty
- BIRADS II: there is some glandular tissue
- BIRADS III: the breast is heterogenously dense

- BIRADS IV: the breast is extremely dense.

Breast cancer is the most fatal cancer type among women in the United States as well as in the European Union [1, 4]. 12% of women develop invasive breast cancer in their lifetime. In a study in 2010, it is estimated that 207090 new cases of invasive breast cancer were expected to be diagnosed alongside 54010 new non-invasive cases. In addition to these numbers, 39840 women in the U.S. were expected to die in 2010 from breast cancer. Pain, skin thickening, nipple discharge, change in breast size or shape are said to be the symptoms of breast cancer [1].

Computer aided diagnosis systems are being developed to assist radiologists, and full automatic medical diagnosis systems are still a hot research topic. Available commercial CAD systems that work on mammographic images try to detect abnormalities in breast tissue automatically. However, some research have shown that there is a high relation between the tissue density and the risk of cancer [5]. When looking for cancer in a breast mammogram, the probability of a false-negative increases with an increase in the density of the parenchymal tissue, because tumors and microcalcifications can hide inside the dense tissue. Moreover, some research have shown that the risk of developing cancer in dense tissue is higher than other tissue types. The positive correlation between increasing breast density with the risk of cancer and the risk of missing a cancerous tissue makes breast tissue density classification an important research topic.

There are several studies focused on classifying breast tissue density in mammographic images. Various local image features such as SIFT features and textons, and global ones such as singular value decomposition (SVD) and histogram based methods, are used in the literature. Generally all proposed methods perform some preprocessing steps, such as contrast enhancement, segmentation and pectoral muscle removal. Mostly supervised classifiers are used to categorize images.

In [5], Bosch et al. presented a method for modeling and classification of breast tissue using local image descriptors and a bag-of-words method. They apply a two-phase preprocessing that consists of segmentation and pectoral muscle extraction. Their segmentation algorithm computes a global gray level histogram of 8 bins.

The minimum value in these 8 values is used to threshold the image. The biggest segment is extracted using a connected component labeling algorithm, and this region is the union of the breast and the pectoral muscle. By using a polynomial modeling approach proposed in [6], they get rid of the pectoral muscle and obtain the region of interest. They compared the performance of different features, namely, textons and SIFT features. Images are represented as bag-of-words, where the visual vocabulary is obtained using a k-means based vector quantization. The distribution of different tissue densities are discovered using probabilistic latent semantic analysis (pLSA). This is an unsupervised method for latent topic discovery in documents, and in this case, the topic is the tissue density. The density distributions are then classified using k-nearest neighbours (kNN) and support vector machine (SVM) classifiers.

In [7], Oliver et al. evaluated different segmentation strategies prior to feature extraction with the aim of density classification. The images are first segmented using four different methods, which are segmentation according to the distance to the skin-line, segmentation through fuzzy c-means, segmentation using fractal analysis and segmentation via statistical analysis. Morphological and texture features are extracted from all of these segmented images and from the images without segmentation. The relative area and the four first histogram moments are used as the morphological features. As texture features, co-occurance matrices are calculated and for each of these matrices, contrast, energy, entropy, correlation, sum average, sum entropy, difference average, difference entropy and homogeneity statistics are used. A Bayesian classifier which combines a k-nearest neighbours (kNN) classifier and C4.5 decision tree is used for classification. The system was tested on the MIAS dataset and results are reported as BIRADS categories.

In [8], Oliveira et al. published their work on tissue density classification in the context of content based image retrieval systems, which employs singular value decomposition (SVD) as the feature extraction method and support vector machines (SVM) for classification. Their purpose of using SVD as the feature extraction method is reducing dimensionality of feature vectors for computational efficiency without losing the textural information.

In [9], Petroudi and Brady proposed a new method based on textons. The novelty they introduced is using a texture descriptor instead of histograms. The texture descriptor evaluates the spatial dependence between the textons that characterize the image. Similar to gray level co-occurance matrices, the texton spatial dependence matrix (TSDM), which can alternatively be called texton co-occurance matrix, is used to capture both structural and statistical information. From different approaches of creating texton maps, in the selected one each texton corresponds to a vector, not to a pixel intensity value or a gradient. The TSDM contains the frequencies or probabilities of texton co-occurances, thus it provides statistical information. Their classification model is built by first creating a texton dictionary from clustered Maximum Response 8 (MR8) filter bank responses of all images per each BIRADS class. Then each pixel in the breast image in the training set is mapped to the closest texton. TSDM matrices for different displacements are computed for each image and each density class is modeled by sets of TSDMs. In order to classify a test image, all steps in the training is applied and the resulting TSDMs are compared to the TSDMs of the learned models. A class assignment is performed using a chi-square test and k-nearest neighbor rule.

In [10] Wang et al. proposed a new method for constructing the visual vocabulary in bag-of-features based classification. The most of the methods using the bag-of-features approach build the vocabulary based on k-means clustering by taking cluster centroids as the visual words. Considering the arguments that the k-means algorithm does not select the most informative words, the researchers proposed jointly using learning and weighting visual words, which they call Joint-ViVo. After segmentation and selecting the ROI, local features, such as local patches and SIFT descriptors, are extracted. By clustering the local features in the training set, the visual vocabulary is built, which consists of cluster centroids as visual words. Histogram of visual word occurances are bag-level representation of images. These histograms are weighted according to their discriminant ability. kNN and SVM classifiers are used for classification of the features.

In this study the aim is to develop an automatic system for classification of mammographic images into different tissue density classes. The MIAS dataset is used for the experiments. In this dataset a full groundtruth is provided which includes

tissue density classes that are fatty, fatty-glandular and dense-glandular [11]. Since the BIRADS annotation for this dataset is not provided, the given class information within the groundtruth is used for the experiments. Local image features are extracted using the Scale-invariant feature transform [12]. A bag-of-features representation is employed and this is used for selecting data samples from the subclasses of each class in order to select the optimal training set. Supervised classification using three different methods, namely Gaussian Mixture Models, Support Vector Machines (SVM) and Learning Vector Quantization (LVQ), is performed in order to learn the dataset and to classify an unknown image.

In the next chapter the system is explained. Theoretical details of the SIFT feature extraction algorithm and the classification methods are given in chapters 3 and 4, respectively. This is followed by the test results and conclusions.

2. SYSTEM OVERVIEW

We propose a system for classifying breast tissue density in mammographic images with the aim of building a part of a bigger system such as computer aided diagnosis (CAD) or content based medical image retrieval (CBMIR) systems. Determining the breast tissue density prior to the tumor detection process is an important procedure in CAD systems working on mammographic images. Similarly, a CBMIR system can take advantage of the density information in indexing which presents the chance of searching over smaller data and thus reducing the processing time.

The proposed system mainly consists of some preprocessing steps, local feature extraction, a training and a test stage. An overview of the system is presented in Figure 2.1. The first three blocks in this figure constitute the preprocessing steps, local feature extraction and classifier design for the training stage, and the last one constitutes the test stage where an unknown image is classified in terms of its tissue density.

We used the MIAS database for experiments. The images in this set are annotated in terms of tissue density type and abnormality type and location if exists. There are three tissue density classes, which are fatty (Class F), fatty-glandular (Class G) and dense-glandular (Class D) [11].

In the first step, the images are cropped to get rid of irrelevant regions that are the pectoral muscle, labels, etc. Then a gray level normalization is applied to images which will then increase the quality of image features. This normalization scales pixel intensities into values between 0 and 255.

Mammographic images have a non-uniform textural content with high variations even in same class of tissue densities. A class may contain images with different visual content and image characteristics such as contrast, brightness, and size and distribution of darker and brighter regions. With the thought that local image features are better for representing such kind of data by modeling the textural content with a minimum

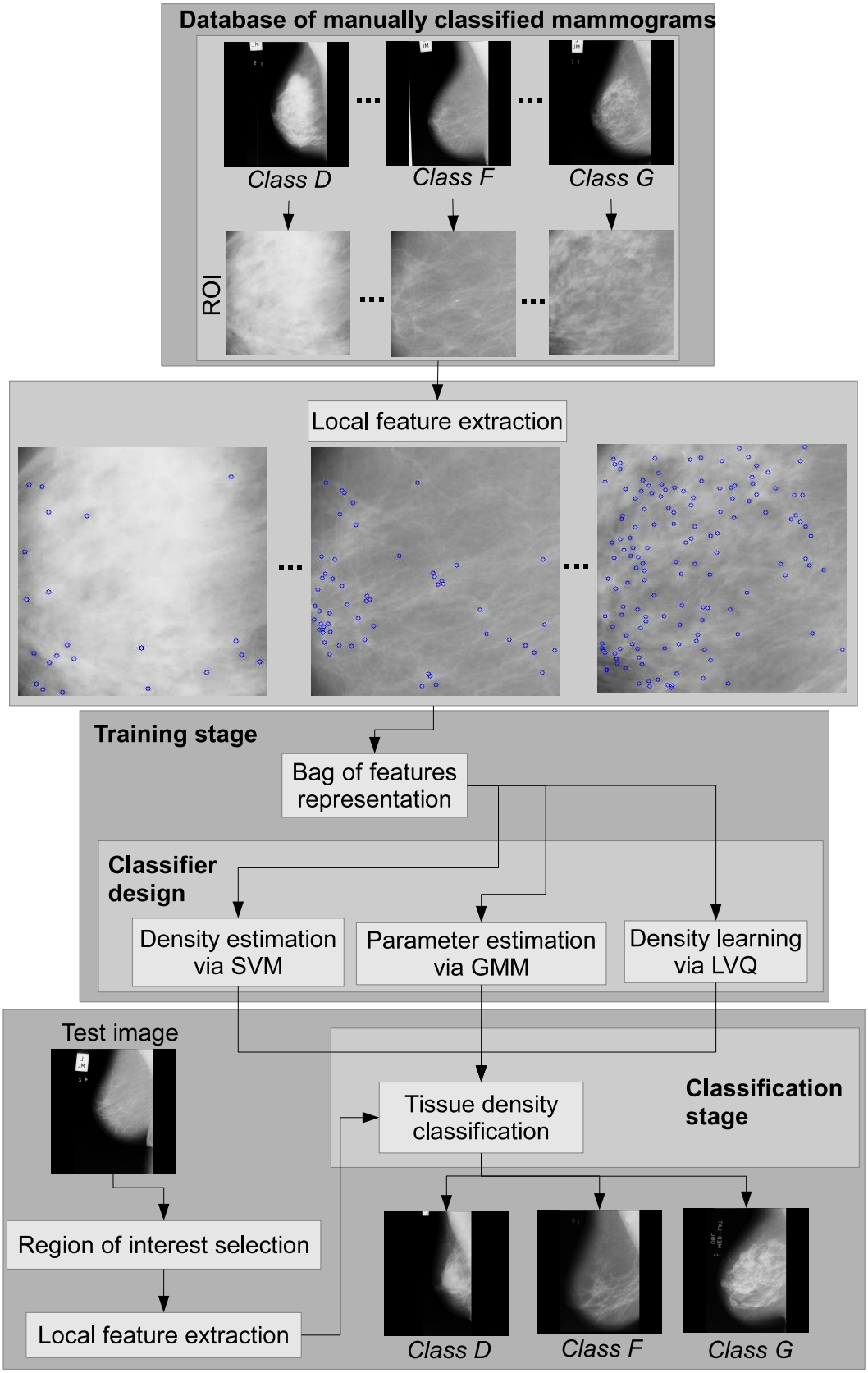


Figure 2.1: System overview.

number of bits, we use scale-invariant feature transform (SIFT) for local feature extraction. SIFT features are commonly used local image features and are extracted based on the gradient histograms of images with a multiscale analysis. Specifically SIFT features are extracted at a number of scales and the region of interest within the mammogram is pointed out with these features named as candidate keypoints. These candidates are better localized, and the ones with low contrast are eliminated. Orientation histograms in a region around these keypoints construct the descriptors [12]. Details of the SIFT algorithm are given in the next chapter. The second block in Figure 2.1 corresponds to the feature extraction step and examples of localized feature points are plotted on mammographic images from different classes. Observe that the density and distribution of descriptors are different for each class.

In the third block of Figure 2.1, first the data is represented using the bag-of-features method with the aim of optimal training set selection. This is a dictionary-based method which is influenced by the bag-of-words method used in document classification where each document is represented by the frequency of words in the vocabulary. Hence the document is called as a bag and the representation is achieved independent of the order of words [13]. In bag-of-features definition the words are the clusters of image features extracted from the visual content [13]. In our problem, we have used bag-of-features approach to specify similar sub-groups of features representing each mammographic tissue density class. As it is known, selection of number of words or number of feature groups included in the dictionary plays the key role in these approaches. We have applied bag-of-features approach by clustering the data into subgroups using the ISODATA algorithm with a large initial number of clusters. Hence the scheme converges to the optimal number of words automatically. By combining the similar clusters, similar data unites into same clusters. The cluster centroids constitute the visual vocabulary. By this method, variations in a class are modeled better, which suits the type of data in mammographic images. For the purpose of optimal training set selection, we construct the training set by taking the vectors from each cluster. This procedure ensures that the training set contains samples from all variations existing in the data.

In order to evaluate the classification performance achieved by the SIFT descriptors, we use three classification methods which are Gaussian mixture models (GMM), support vector machines (SVM) and learning vector quantization (LVQ). These algorithms are placed in the third block in Figure 2.1.

The characteristics of the data make it suitable to be modeled by Gaussian Mixture Models (GMM). This is a parametric method where a mixture of Gaussians is fit to each class by estimating the parameters of the mixture distribution [14]. The idea behind using a mixture instead of a single Gaussian is that it is better to represent each sub-population (word) in the data by an individual Gaussian distribution, thus the mixture can model the entire data (bags-of-features) well.

Support vector machines are used as a second classifier. SVM is a statistical method and it can be used for supervised binary classification problems [14]. By using a kernel function, the linearly non-seperable data can be mapped into another feature space where a maximum-margin hyperplane can be fit between the classes. Unlike the GMM, the SVM enables us to estimate the probability density function representing each class. Hence by using the SVM we have examined the mammogram classification performance with a supervised nonparametric statistical classifer rather than a parametric classifier as in the GMM.

As the third option in the classifier design, learning vector quantization algorithm is used [15]. The LVQ is a learning algorithm with similarities and relations to self-organizing maps and vector quantization except being a supervised algorithm. This method models the data by a reward-punishment scheme. The data is split into clusters, and each cluster is represented by a number of codevectors which are the components of a codebook. This method is suitable for our problem since the data is examined in detail by representing the sub-groups by different sets of codevectors. Unlike the first two classifiers, the LVQ does not estimate the density function instead applies a k-nearest neighbor type classification scheme.

Classification of a new observation referred to as the test image is performed within the test module which constitutes a separate block in the system, as can be seen in Figure 2.1. Cropping, normalization and local feature extraction operations are applied to

the test image as in the first two blocks of the training stage. Each feature of the test image is classified using one of the classifiers which were designed in the training stage. A decision is made by majority voting of class labels assigned to the features of the image and the decision is reported as one class label for the image, namely, fatty, fatty-glandular or dense-glandular.

3. EXTRACTION OF LOCAL IMAGE FEATURES

In this study the problem is classifying the tissue density in mammographic images. These images have a textural content and images from different classes may have similar global image characteristics such as contrast and brightness as well as images from the same classes may have different characteristics. These reasons led us to use local image features in order to utilize the texture and capture the local characteristics.

We use Scale-Invariant Feature Transform (SIFT) for feature extraction. SIFT provides local image features that are invariant to scaling, rotation and partially robust to affine distortion, change in 3D viewpoint, addition of noise and change in illumination [12]. This algorithm is widely used for image matching and object recognition purposes.

Several features can be extracted from different locations in a single image using SIFT algorithm. Each feature is represented by four floating point numbers and a descriptor vector of 128 integers. Using a cascade filtering approach, operations with high complexity are applied only to image parts that pass the initial elimination steps.

SIFT algorithm searches for potential interest points in different scales of the image. This approach allows selecting the interest points that exist in different scales, thus provides scale-invariance of the features.

There are four main operations in SIFT algorithm. First a scale-space is generated and extremum points are found using the difference-of-Gaussians (DoG) approach. Interest point candidates which are obtained in this step are then better localized both spatially and between scales, and in the meantime a keypoint selection/elimination procedure is performed. An orientation is assigned to each keypoint. Finally, using orientation histograms, a descriptor is computed for each keypoint. In the rest of this section these operations are discussed in detail.

In order to generate the scale-space, 2D-convolution of the image and a Gaussian function with variable standard deviation is calculated. This is formulated as follows:

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \tag{3.1}$$

In (**3.1**), $L(x,y,\sigma)$ is the scale space, $G(x,y,\sigma)$ is the variable-scale Gaussian function, $I(x,y)$ is the input image and $*$ is the 2D-convolution operator. $G(x,y,\sigma)$ is defined as follows:

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{3.2}$$

Stable keypoint candidates can be located in scale-space by convolving the scale-space extrema obtained using the difference-of-Gaussians function with the image [12]. Difference-of-Gaussians function is defined as $D(x,y,\sigma)$, and it can be computed as follows:

$$\begin{aligned} D(x,y,\sigma) &= (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \\ &= L(x,y,k\sigma) - L(x,y,\sigma) \end{aligned} \tag{3.3}$$

In (**3.3**) difference of two adjacent scales, which are separated by a multiplicative constant factor k, is convolved with the image. This approach provides efficient computation since difference-of-Gaussians can be obtained just by subtracting two smoothed images. Furthermore, difference-of-Gaussians function is a close approximation to the scale-normalized Laplacian of Gaussian function which is defined as $\sigma^2\nabla^2 G$. Various studies have shown that scale-invariance requires a normalization of the Laplacian by a factor, σ^2, and that when compared to some image functions such as the gradient, Hessian and Harris corner function, the extrema of the scale-normalized Laplacian of Gaussian function provides the most stable features.

Laplacian of Gaussian can be written in a form similar to the heat diffusion equation as in the following:

$$\frac{\partial G}{\partial \sigma} = \sigma\nabla^2 G \tag{3.4}$$

Using the finite difference approximation method, (**3.4**) can be written as follows:

$$\sigma\nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x,y,k\sigma) - G(x,y,\sigma)}{k\sigma - \sigma} \tag{3.5}$$

Using (**3.5**), difference of two nearby Gaussians separated by a scaling factor of k can be written as follows:

$$G(x,y,k\sigma) - G(x,y,\sigma) \approx (k-1)\sigma^2\nabla^2 G \tag{3.6}$$

Another efficiency that difference-of-Gaussians provides is that it already includes the normalization factor σ^2 which is needed for the scale-invariance, and this can be seen in (**3.6**).

In practice, a scale-space is constructed by convolving the image with Gaussians of variable-scale, and this group of smoothed images is called an octave. The second octave is obtained by downsampling each image in the first octave. By subtracting the adjacent images, difference-of-Gaussians for each octave are obtained. This procedure is highly efficient in computation.

Extrema of the difference-of-Gaussians is detected by comparing each sample point of a smoothed image to its eight neighbors in the current image and nine neighbors in the two neighbor scales above and below the current image. If the sample point is the largest or the smallest one in all these twenty six comparisons, it is considered as an extremum.

The second major step in SIFT algorithm is localizing keypoints in more detail. This procedure also provides elimination of keypoint candidates with low contrast which are sensitive to noise. In order to find the accurate location of an extremum point, an interpolation is performed by fitting a 3D quadratic function to local sample points. For this purpose the scale-space $D(x,y,\sigma)$ is shifted to locate the origin at the sample point. Then Taylor expansion of $D(x,y,\sigma)$ up to quadratic terms is applied as follows:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \tag{3.7}$$

In (**3.7**), $\mathbf{x} = (x,y,\sigma)^T$ is the offset from the sample point. Let $\hat{\mathbf{x}}$ be the location of the extremum. Taking the derivative of the function with respect to $\mathbf{x}$ in (**3.7**) and setting it to zero gives the following equation:

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x^2}}^{-1}\frac{\partial D}{\partial \mathbf{x}} \tag{3.8}$$

Differences of neighboring sample points can be used as approximates of the Hessian and derivative of D. If the offset $\hat{\mathbf{x}}$ has a value bigger than 0.5 in any dimension, the

sample point is changed. The interpolated location of the extremum is obtained by adding $\hat{\mathbf{x}}$ to the location of the sample point.

To eliminate the extrema with low contrast which are unstable because of being sensitive to noise, equations (**3.7**) and (**3.8**) are used together giving the following equation:

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2}\frac{\partial D^T}{\partial \mathbf{x}}\hat{\mathbf{x}} \tag{3.9}$$

Sample points with low $\mid D(\hat{\mathbf{x}}) \mid$ values are considered as unstable extrema, thus, are rejected.

Another procedure to reject unstable extrema is eliminating the sample points that are localized along the edges in the image. On these locations the difference-of-Gaussians function has a strong response which causes it to be sensitive to noise. For this purpose, 2x2 Hessian matrix at the location and scale of the keypoint is calculated since the eigenvalues of this matrix are proportional to the principal curvatures of the difference-of-Gaussians function. The principal curvatures of a poorly defined peak in the difference-of-Gaussians function get large values across the edge and small values in the perpendicular direction. The Hessian matrix is defined as follows:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{3.10}$$

Differences of neighboring sample points can be used as approximates to the derivatives in the Hessian matrix. Assuming that α and β are the two eigenvalues of H with the largest and smallest magnitudes, respectively, the sum and product of these eigenvalues can be computed using the trace and determinant of H, respectively, as in the following equations:

$$\begin{aligned} Tr(H) &= D_{xx} + D_{yy} = \alpha + \beta \\ Det(H) &= D_{xx}D_{yy} - D_{xy}D_{xy} = \alpha\beta \end{aligned} \tag{3.11}$$

Defining $r = \alpha/\beta$ and using (**3.11**), the following equation can be written:

$$\frac{Tr(H)^2}{Det(H)^2} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \tag{3.12}$$

From (**3.12**) it is seen that instead of using magnitudes of the eigenvalues, their ratio can be used for eliminating principal curvatures with large values. The keypoints which

do not satisfy the following condition are eliminated:

$$\frac{Tr(H)^2}{Det(H)^2} < \frac{(r+1)^2}{r} \tag{3.13}$$

The third main step in SIFT is the orientation assignment. This is performed in order to obtain rotation invariant features by representing the keypoint descriptors relative to their own orientations. The Gaussian smoothed image, L, in the closest scale to the scale of the keypoint is used in this procedure. A gradient magnitude $m(x,y)$ and orientation $\theta(x,y)$ are computed using pixel differences for each image sample $L(x,y)$ as formulated in the following:

$$\begin{aligned} m(x,y) &= \sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2} \\ \theta(x,y) &= \arctan\left(\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)}\right) \end{aligned} \tag{3.14}$$

Using the gradient orientations assigned to sample points within a region around the keypoint, a histogram with 36 bins is generated. Prior to the histogram forming, each orientation is weighted by its magnitude and a Gaussian weighted circular window. The highest peak in the histogram, and the others within the 80% of the highest one are used to create keypoints with those orientations, thus there may be keypoints with same location and scale but different orientations. For better accuracy, a parabola is fit to the three peaks closest to each peak.

The fourth and last major step in SIFT feature extraction is the descriptor assignment to each keypoint. In a region around the keypoint, the gradient magnitudes and orientations are calculated for each image sample and the magnitudes are weighted by a circular Gaussian window with a scale selected using the scale of the keypoint. The coordinates of the descriptor and the gradient orientations are rotated relative to the keypoint orientation and this procedure provides orientation invariance. The objective in weighting the gradient magnitudes using the Gaussian window is to give more significance to the gradients that are closer to the descriptor center. Orientation histograms with eight bins are formed using the weighted orientations and their magnitudes. A peak in this histogram is the sum of the magnitudes of sample points whose orientations are members of that bin. Each magnitude in a bin is weighted by $1-d$, where d is the distance of the sample from the central value of the bin.

The descriptor vector is formed using the peaks in the histograms in a 4x4 array, each of which is formed within a 4x4 region of sample points. Since each histogram has eight bins, the descriptor vector has 4x4x8=128 elements. This vector is then normalized to unit length to reduce the effects of changes in illumination. Elements of this vector is thresholded and renormalized to unit length in order to reduce the effects of large gradient magnitudes which are caused by non-linear illumination changes.

4. CLASSIFIER DESIGN

Mammographic images have a non-uniform textural content. In this study we have used the MIAS dataset in which the images are categorized into three tissue density classes that are fatty, fatty-glandular and dense-glandular [11]. The challenge in this problem is that some images from different classes are similar in terms of image characteristics such as contrast and brightness, and their visual content; and there is a high variation in each class, in other words, images from the same class may look different and may have very different image characteristics.

As explained in the previous chapters, we use local image features in order to capture the in-class variations and local image characteristics. We use scale-invariant feature transform (SIFT) [12] as the local feature extraction algorithm. The extracted features are represented by a bag-of-features scheme.

We have evaluated three different supervised classification algorithms which are classification via Gaussian mixture models, support vector machines and learning vector quantization. These algorithms are explained in detail in the following sections.

4.1 Classification Using Gaussian Mixture Models

Gaussian mixture models (GMM) are a special case of statistical mixture models where the model is built by a number of multivariate Gaussian distributions. In this method a parametric probability density function that is the sum of weighted Gaussian densities is fit to the data by estimating the parameters of the model [14, 16].

In our problem there are three tissue density classes which may sometimes have similar image characteristics. The textural structure of the mammographic images are hard to model and there is a high variation in each class. In order to model these variations, Gaussian mixture models may be useful since this algorithm tries to fit several Gaussian distributions over different parts of the data.

A Gaussian mixture model is parametrized by the mean vectors, covariance matrices and mixture weights of each component. These parameters are estimated and a mixture model is built to fit the observations in a class.

In the training phase, a model for each class is built separately by estimating its parameters using the training data of that class. Each model may include different number of Gaussians. In our work we modeled each class by using a mixture of eight Gaussians distributions.

Maximum likelihood (ML) estimation is a method that can be used to estimate the parameters of the Gaussian mixture models by finding the parameters which maximize the likelihood of the models.

The form of a finite mixture model is as follows:

$$p(x) = \sum_{j=1}^{g} \pi_j p(x;\theta_j) \tag{4.1}$$

where g is the number of mixture components, π_j are the mixing proportions, and $p(x;\theta_j)$ are the component density functions for $j = 1,...,g$. The mixing proportions, π_j, sum up to 1 and $\pi_j \geq 0$. When the mixture components are Gaussians, $p(x;\theta_j)$ are multivariate Gaussian distributions and the paremeter set of a distribution is $\theta_j = \{\mu_j, \Sigma_j\}$.

For a set of n observations, $(x_1,...,x_n)$, and a set of parameters, $\Psi = \{\pi_1,...,\pi_g;\theta_1,...,\theta_g\}$, the likelihood function of the mixture model can be written as follows:

$$L(\Psi) = \prod_{i=1}^{n} \sum_{j=1}^{g} \pi_j p(x_j \mid \theta_j) \tag{4.2}$$

where $p(x \mid \theta_j)$ are the component densities with dependence on their parameters.

Generally it is impossible to differentiate L with respect to Ψ, thus the expectation-maximization (EM) algorithm can be used iteratively to solve this. EM algorithm was proposed by Dempster et al. in 1977 for missing data estimation.

Let $y^T = (x^T, z^T)$ be the complete data that includes class labels, where z is a vector of class labels which includes 1 in the k^{th} position if x belongs to the category k and zero

elsewhere. The likelihood of y is given as follows:

$$\begin{aligned} g(y \mid \Psi) &= p(x \mid z, \Psi) p(z \mid \Psi) \\ &= p(x \mid \theta_k) \pi_k \end{aligned} \tag{4.3}$$

and this can be written as

$$g(y \mid \Psi) = \prod_{j=1}^{g} \left[p(x \mid \theta_j) \pi_j \right]^{z_j} \tag{4.4}$$

since z_j is 1 only if $j = k$.

The likelihood of x, which is the mixture distribution, can be written as follows:

$$\begin{aligned} p(x \mid \Psi) &= \sum_{\text{all possible values of} z} g(y \mid \Psi) \\ &= \sum_{j=1}^{g} \pi_j p(x \mid \theta_j) \end{aligned} \tag{4.5}$$

For n observations, the following likelihood can be written:

$$g(y_1, ..., y_n \mid \Psi) = \prod_{i=1}^{n} \prod_{j=1}^{g} \left[p(x_i \mid \theta_j) \pi_j \right]^{z_{ji}} \tag{4.6}$$

and by taking the logarithm of this, the following is obtained:

$$\log(g(y_1, ..., y_n \mid \Psi)) = \sum_{i=1}^{n} z_i^T l + \sum_{i=1}^{n} z_i^T u_i(\theta) \tag{4.7}$$

where the vector l contains j^{th} component $\log(\pi_j)$, u_i contains j^{th} component $\log(p(x_i \mid \theta_j))$ and z_i contains z_{ji}, $j = 1, ..., g$, where z_{ji} are class labels which are 1 if x_i belongs to the group j, and 0 otherwise.

Each iteration in EM algorithm consists of an expectation (E) step, and a maximization (M) step.

For the E-step, the following equation is formed:

$$Q(\Psi, \Psi^{(m)}) = \sum_{i=1}^{n} w_i^T l + \sum_{i=1}^{n} w_i^T u_i(\theta_i) \tag{4.8}$$

where $w_i = \mathrm{E}[z_i \mid x_i, \Psi^{(m)}]$, with j^{th} component, the probability that x_i belongs to the group j given the current estimates $\Psi^{(m)}$, given by the following:

$$w_{ij} = \frac{\pi_i^{(m)} p(x_i \mid \theta_j^{(m)})}{\sum_k \pi_k^{(m)} p(x_i \mid \theta_k^{(m)})} \tag{4.9}$$

The maximization step maximizes Q with respect to Ψ. The parameters π_i and θ_i are maximized in turn. Maximizing Q with respect to π_i is performed by differentiating $Q - \lambda\left(\sum_{j=1}^{g}\pi_j - 1\right)$ with respect to π_j, where λ is a Lagrange multiplier and this gives the following equation:

$$\sum_{i=1}^{n} w_{ij}\frac{1}{\pi_j} - \lambda = 0 \tag{4.10}$$

The constraint $\sum\pi_j = 1$ gives $\lambda = \sum_{j=1}^{g}\sum_{i=1}^{n} w_{ij} = n$, and the estimate of π_j can be written as follows:

$$\hat{\pi}_j = \frac{1}{n}\sum_{i=1}^{n} w_{ij} \tag{4.11}$$

Since $\theta_j = (\mu_j, \Sigma_j)$ for Gaussian mixture models, the mean and covariance matrix must be estimated separately. Estimate of the mean vector is obtained by differentiating Q with respect to μ_j and equating to zero as in the following:

$$\sum_{i=1}^{n} w_{ij}(x_i - \mu_j) = 0 \tag{4.12}$$

and the estimate of μ_j is found as follows:

$$\hat{\mu}_j = \frac{\sum_{i=1}^{n} w_{ij}x_i}{\sum_{i=1}^{n} w_{ij}} = \frac{1}{n\hat{\pi}_j}\sum_{i=1}^{n} w_{ij}x_i \tag{4.13}$$

In order to estimate the covariance matrix, Q is differentiated with respect to Σ_j and equated to zero, which gives the following update equation:

$$\begin{aligned}\hat{\Sigma}_j &= \frac{\sum_{i=1}^{n} w_{ij}(x_i - \hat{\mu}_j)(x_i - \hat{\mu}_j)^T}{\sum_{i=1}^{n} w_{ij}} \\ &= \frac{1}{n\hat{\pi}_j}\sum_{i=1}^{n} w_{ij}(x_i - \hat{\mu}_j)(x_i - \hat{\mu}_j)^T\end{aligned} \tag{4.14}$$

The next iteration starts using this set as the initial, and this procedure is repeated until convergence to a local maximum of the likelihood function is reached.

4.1.1 Classification of observed mammograms

In order to predict the classes which a set of sample vectors belong to, probability distribution functions of each model are evaluated at each sample vector. Let there be C classes with the estimated parameters $\Psi_1, ..., \Psi_C$, and s patterns to be classified,

$x_1, ..., x_s$. The likelihood of the observation given each class model, $L(x_i \mid \Psi_j)$, is calculated for $i = 1, ..., s$ and $j = 1, ..., C$. Therefore, for each observation we assign a class label by ML estimation with the following rule

$$c_i = \arg\max_j \left\{ L(x_i \mid \Psi_j) \right\} \tag{4.15}$$

In our problem, there are three classes (C=3) which are fatty, fatty-glandular and dense-glandular.

4.2 Classification Using Support Vector Machines

Support vector machines are non-probabilistic supervised classifiers which look for the maximal margin hyperplane, which separates the data best by leaving a gap between the two classes. By mapping the data vectors to a high dimensional feature space, support vector machines can be applied to linearly non-separable data [14].

In this section classification using support vector machines will be described. The model for two linearly separable classes will be given first, and then this will be extended to linearly non-separable data.

4.2.1 Classification of linearly separable classes

Let x_i be the feature vectors of the training set X, where $i = 1, 2, ..., n$, and each input x_i has D attributes. These vectors belong to either of two linearly separable classes, ω_1 and ω_2 with labels $y_i = \pm 1$. Here linearly separable means that a line can be drawn between the classes when $D = 2$, and when $D > 2$, these classes can be separated by a hyperplane. The training set is given as $\{x_i, y_i\}$ pairs.

The discriminant function which is used to classify all of the training vectors is described as follows:

$$g(x) = w^T \cdot x + w_0 \tag{4.16}$$

The decision rule for this discriminant function is given as follows:

$$w^T x + w_0 \begin{cases} > 0 \Rightarrow x \in \omega_1, & y_i = +1 \\ < 0 \Rightarrow x \in \omega_2, & y_i = -1 \end{cases} \tag{4.17}$$

All the observations in the training set can be correctly classified if the following condition is assured [14]:

$$y_i(w^T x_i + w_0) > 0 \text{ for all } i \tag{4.18}$$

There may be more than one separating hyperplanes. The maximal margin classifier chooses the hyperplane with leaving the maximum margin from both sides. It is assumed that a larger margin will give a small generalization error [14].

A margin $b > 0$ which satisfies the following condition is needed:

$$y_i(w^T + w_0) \geq b \tag{4.19}$$

A solution for which all x_i vectors are at a distance greater than $b/ \mid w \mid$ can be found, and when b, w_0 and w are scaled, the condition in (**4.19**) is still satisfied. Assuming $b = 1$, the canonical hyperplanes can be defined as follows:

$$\begin{aligned} H_1 &: w^T x + w_0 = +1 \\ H_2 &: w^T x + w_0 = -1 \end{aligned} \tag{4.20}$$

With these hyperplanes, the decision rules become as in the following:

$$\begin{aligned} w^T x_i + w_0 &\geq +1 \text{ for } y_i = +1 \\ w^T x_i + w_0 &\leq -1 \text{ for } y_i = -1 \end{aligned} \tag{4.21}$$

There is the distance, $1/ \mid w \mid$, between each of these hyperplanes and the separating hyperplane, $g(x) = 0$, and this distance is called the margin. The observations closest to the separating hyperplane, which lie on the canonical hyperplanes, are called the support vectors.

Maximizing the margin means minimizing $\mid w \mid$ with the following constraints:

$$\text{C1} : y_i(w^T x_i + w_0) \geq 1, \; i = 1, ..., n \tag{4.22}$$

Objective function for this optimization problem is given as follows:

$$L_p = \frac{1}{2} w^T w - \sum_{i=1}^{n} \alpha_i (y_i(w^T x_i + w_0) - 1) \tag{4.23}$$

where $\{\alpha_i, i = 1, ..., n; \alpha_i \geq 0\}$ are the Lagrange multipliers.

Finding the saddlepoint of the objective function L_p is equivalent to minimizing $w^T w$ subject to the constraints in (**4.22**). When finding the saddlepoint, L_p is minimized with respect to w and w_0 and maximized with respect to α_i. Differentiating L_p with respect to w and w_0 and equating it to zero gives the following:

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$
$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{4.24}$$

With the objective function L_p, this gives

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{4.25}$$

which is the dual form of the Lagrangian and this is maximized with respect to α_i subject to

$$\alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{4.26}$$

In (**4.25**) the dual variables are the Lagrange multipliers, α_i, and the number of parameters equal to the number of patterns, n.

Data points with non-zero Lagrange multiplier are the support vectors since they lie on the canonical hyperplanes.

4.2.1.1 Classification of a new observation

Having the Lagrange multipliers, w_0 can be found using any of the support vectors from the equation

$$\alpha_i (y_i (x_i^T w + w_0) - 1) = 0 \tag{4.27}$$

or using the average of all support vectors from the equation

$$n_{sv} w_0 + w^T \sum_{i \in SV} x_i = \sum_{i \in SV} y_i \tag{4.28}$$

where n_{sv} is the number of support vectors and SV is the set of support vectors. w can be found by the following:

$$w = \sum_{i \in SV} \alpha_i y_i x_i \tag{4.29}$$

In order to classify a new observation, x, the sign of $w^T x + w_0$ is used. The decision rule is as follows:

$$\text{assign } x \text{ to } \omega_1 \text{ if}$$
$$\sum_{i \in SV} \alpha_i y_i x_i^T x - \frac{1}{n_{SV}} \sum_{i \in SV} \sum_{j \in SV} \alpha_i y_i x_i^T x_j + \frac{1}{n_{SV}} \sum_{i \in SV} y_i > 0 \tag{4.30}$$

4.2.2 Classification of linearly non-separable data

In general there is no linear boundary between classes, so a hyperplane for separating these classes cannot be found. When data is not linearly separable, the support vector algorithm may be applied in a transformed feature space in which data becomes linearly separable. Let $\phi(x)$ be the transformed feature space. The discriminant function can be written as follows:

$$g(x) = w^T \phi(x) + w_0 \tag{4.31}$$

The following equation is the decision rule for the discriminant function in (**4.31**).

$$w^T \phi(x) + w_0 \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow x \in \begin{cases} \omega_1, y_i = +1 \\ \omega_2, y_i = -1 \end{cases} \tag{4.32}$$

The maximum margin can be found by maximizing the Lagrangian. The dual form of the Lagrangian can be written as follows:

$$L_D = \sum_{i=1}^{n} \alpha_i \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \phi^T(x_i) \phi^T(x_j) \tag{4.33}$$

where $y_i = \pm 1, i = 1, ..., n$ are the class indicators and $\alpha_i, i = 1, ..., n$ are the Lagrange multipliers which satisfy the following for a regularization parameter C:

$$0 \leq \alpha_i \leq C$$
$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{4.34}$$

The support vectors may be found by maximizing (**4.33**) under the constraints given in (**4.34**) and selecting the non-zero values of α_i. w can be written as follows:

$$w = \sum_{i \in SV} \alpha_i y_i \phi(x_i) \tag{4.35}$$

In order to classify a new observation, x, the sign of the following is assigned to x as the class label:

$$g(x) = \sum_{i \in SV} \alpha_i y_i \phi^T(x) + w_0 \tag{4.36}$$

where

$$w_0 = \frac{1}{N_{\widehat{SV}}} \left\{ \sum_{i \in \widehat{SV}} y_i - \sum_{i \in SV, j \in \widehat{SV}} \alpha_i y_i \phi^T(x_i) \phi(x_j) \right. \tag{4.37}$$

In (**4.37**), SV is the set of support vectors with associated values of $alpha_i$ satisfying $0 < \alpha_i \leq C$ and $\widehat{SV}$ is the set of $N_{\widehat{SV}}$ support vectors satisfying $0 < \alpha_i < C$.

A kernel function can be used instead of the scalar products between transformed feature vectors which the optimization of the dual form given in (**4.33**) and classification of a sample given in (**4.36**) rely on. This kernel function can be written as follows:

$$K(x,y) \phi^T(x) \phi(y) \tag{4.38}$$

Using this kernel function, there is no need to know ϕ explicitly and the discrimination function can be written as in the following:

$$g(x) = \sum_{i \in SV} \alpha_i y_i K(x_i, x) + w_0 \tag{4.39}$$

The kernels used with support vector machines are given in the following:

$$\text{Polynomial: } (1 + x^T y)^d$$

$$\text{Radial Basis: } \exp(- \mid x - y \mid^2 / \sigma^2)$$

$$\text{Sigmoid: } \tanh(k x^T y - \delta)$$

4.3 Classification Using Learning Vector Quantization

Learning vector quantization is a statistical classification method which has a strong relation and similarity to vector quantization and self-organizing maps. Its main difference from these two methods is that it is a supervised learning method while the other two are unsupervised methods [15].

In our problem, since the data have in-class variations, learning vector quantization can be used because this algorithm learns the sub-groups in the data iteratively by a reward-punishment scheme.

In order to define the discriminant functions in relation to the Bayes theory, let all x vectors be derived from a finite set of classes $\{S_k\}$. $P(S_k)$ is the a priori probability of

class S_k and $p(x \mid x \in S_k)$ is the class-conditional probability density function of x on S_k. Under these assumptions, the discriminant functions are defined as follows:

$$g_k(x) = p(x \mid x \in S_k)P(S_k) \tag{4.40}$$

Optimal classification of x samples, which means that the rate of misclassifications is minimized, is obtained when the discriminant function gets its highest value, as formulated in the following:

$$g_c(x) = \max_k \{g_k(x)\} \tag{4.41}$$

where the sample x belongs to the class S_c.

In learning vector quantization, a subset of codebook vectors is assigned to each class S_k. To classify an observation vector x, the codebook vector m_i with the minimum Euclidean distance to x is selected and it is decided that x belongs to the same class as m_i. Even if the class distributions overlap, it is possible to select the codebook vectors by placing them without overlapping in feature space. In this method, it is more important to select the vector m_i that minimizes the average expected misclassification probability in the nearest neighbor rule used for classification.

4.3.1 LVQ1 algorithm

Let the training data consist of x observation vectors, and several codebook vectors, m_i, are assigned to each class of these sample vectors. It is assumed that the observation vector x belongs to the same class as its closest codebook vector. The index of the closest codebook vector m_i to x is defined as follows:

$$c = \arg\min_i \{\|x - m_i\|\} \tag{4.42}$$

In (**4.42**), c is the index of the winner codebook vector from the list of all codebook vectors. Assuming x is a natural, stochastic, continuous-valued vector, the probability that more than one minima occurs is zero. In other words, there is only one winner codebook vector for an observation.

As the learning process, the winner codebook vector is updated iteratively starting with properly initialized values. Update rules for the LVQ1 algorithm is given in (**4.43**).

Here x^t is the sample vector and m_i^t is the closest codebook vector at iteration t. α^t is called the learning rate, and it takes values between 0 and 1. Generally α^t is decreased in time, and its initial value should be smaller than 0.1.

$$\begin{gathered}\text{if } x \text{ and } m_c \text{ belong to the same class, } m_c^{t+1} = m_c^t + \alpha^t[x^t - m_c^t] \\ \text{if } x \text{ and } m_c \text{ belong to different classes, } m_c^{t+1} = m_c^t - \alpha^t[x^t - m_c^t] \\ \text{for } i \neq c,\ m_i^{t+1} = m_i^t \end{gathered} \tag{4.43}$$

Vector quantization approximates the probability density function of the observation vector x, $p(x)$, or a monotonic function of it. Considering approximation to a non-negative density function $f(x)$, let the Bayesian borders be defined by the discriminant functions in (**4.40**) and (**4.41**), and the borders obtained using $f(x)$ be defined by $f(x) = 0$ where x belongs to the class B_k, $h \neq k$, and

$$f(y) = p(x \mid x \in S_k)P(S_k) - \max_h \{p(x \mid x \in S_h)P(S_h)\} \tag{4.44}$$

Optimal Bayesian borders can be seen in Figure 4.1(a) in the case of scalar observations from three classes, S_1, S_2 and S_3, which are defined by their distributions $p(x \mid x \in S_k)P(S_k)$ in the horizontal axis. Borders are shown by dotted lines. The function defined in (**4.44**) is shown in Figure 4.1(b), and it has zero values at the Bayesian borders shown in Figure 4.1(a) [15]. When the point density of the m_i that approximates $f(x)$ is defined using vector quantization, this density also has zero values at all Bayesian borders. With enough number of codebook vectors, $f(x)$ defined in (**4.44**) and vector quantization together define the Bayesian borders with good accuracy [15].

4.3.2 The Optimized-Learning-Rate LVQ1 (OLVQ1) algorithm

In the OLVQ1 algorithm, an individual learning-rate factor α_i^t is assigned to each winner codebook vector m_i. Update equations in (**4.43**) become

$$\begin{gathered}\text{if } x \text{ is classified correctly, } m_c^{t+1} = m_c^t + \alpha_c^t[x^t - m_c^t] \\ \text{if } x \text{ is not classified correctly, } m_c^{t+1} = m_c^t - \alpha_c^t[x^t - m_c^t] \\ \text{for } i \neq c,\ m_i^{t+1} = m_i^t \end{gathered} \tag{4.45}$$

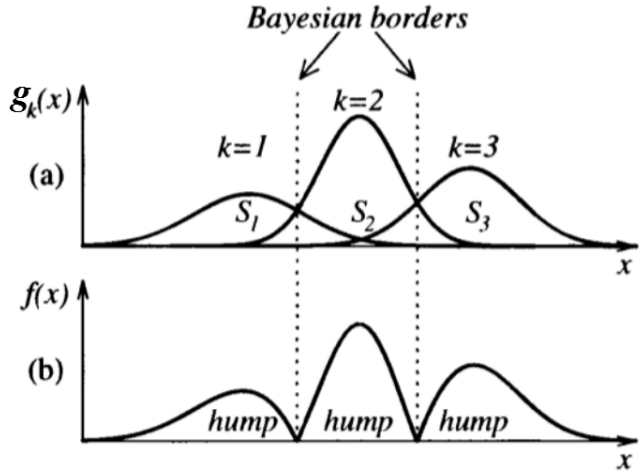


Figure 4.1: (a) Optimal Bayesian borders (b) Another non-negative density function

By defining $s^t = +1$ if the classification is correct and $s^t = -1$ if the classification is incorrect, (**4.45**) can be written as follows:

$$m_c^{t+1} = [1 - s^t \alpha_c^t] m_c^t + s^t \alpha_c^t x^t \tag{4.46}$$

m_c^{t+1} contains a trace of x^t through the last term in (**4.46**), and traces of the earlier $x^{t'}, t' = 1, 2, ..., t-1$ through m_c^t. In an iteration, the magnitude of the last trace of x^t is scaled down by the factor α_c^t. In the same iteration, the trace of x^{t-1} is scaled by $[1 - s^t \alpha_c^t] \alpha_c^{t-1}$. Since these two scaling factors are identical, the following equation can be written:

$$\alpha_c^t = [1 - s^t \alpha_c^t] \alpha_c^{t-1} \tag{4.47}$$

If this condition holds for all t, it can be shown that the traces collected up to iteration t of all the earlier $x^{t'}$ will be scaled down equally; thus the optimal values of α_i^t can be determined as follows:

$$\alpha_c^t = \frac{\alpha_c^{t-1}}{1 + s^t \alpha_c^{t-1}} \tag{4.48}$$

4.3.3 LVQ2.1 algorithm

In this algorithm the two closest codebook vectors, m_i and m_j, one of which belongs to the correct class and the other belongs to a wrong one, are updated. The observation vector must be in a zone called window, which is defined around the midplane of the two closest codebook vectors. Observation x is in the window when the following

condition is satisfied:

$$min\left(\frac{d_i}{d_j}, \frac{d_j}{d_i}\right) > s \text{ where } s = \frac{1-w}{1+w} \tag{4.49}$$

In (**4.49**), w is the relative width of the window, and d_i and d_j are the Euclidean distances of x from m_i and m_j, respectively. A recommended width for the window is between 0.2 and 0.3.

Similar to (**4.43**), update equations for the LVQ2.1 algorithm are as follows:

$$\begin{aligned} m_i^{t+1} &= m_i^t - \alpha^t[x^t - m_i^t] \\ m_j^{t+1} &= m_j^t + \alpha^t[x^t - m_j^t] \end{aligned} \tag{4.50}$$

In (**4.50**), m_i and m_j are the two closest codebook vectors to the observation x. m_j belongs to the same class as x, and m_i belongs to an other class.

LVQ2.1 algorithm is the same as LVQ2 with an improvement that either m_i or m_j can be the closest codebook vector.

4.3.4 Multi-pass LVQ

This setting is used to speed-up the learning process. First it uses OLVQ1 algorithm, and then a long fine tuning pass is made using one of LVQ1, LVQ2.1 or LVQ3 algorithms [17].

4.3.5 Hierarchical LVQ

In this method, first a model is constructed using an LVQ algorithm. The obtained codebook vectors are considered as cluster centroids, and sub-models are constructed under each cluster. The sub-models that outperform their parent codebook vectors are kept as parts of the model [17].

5. TEST RESULTS

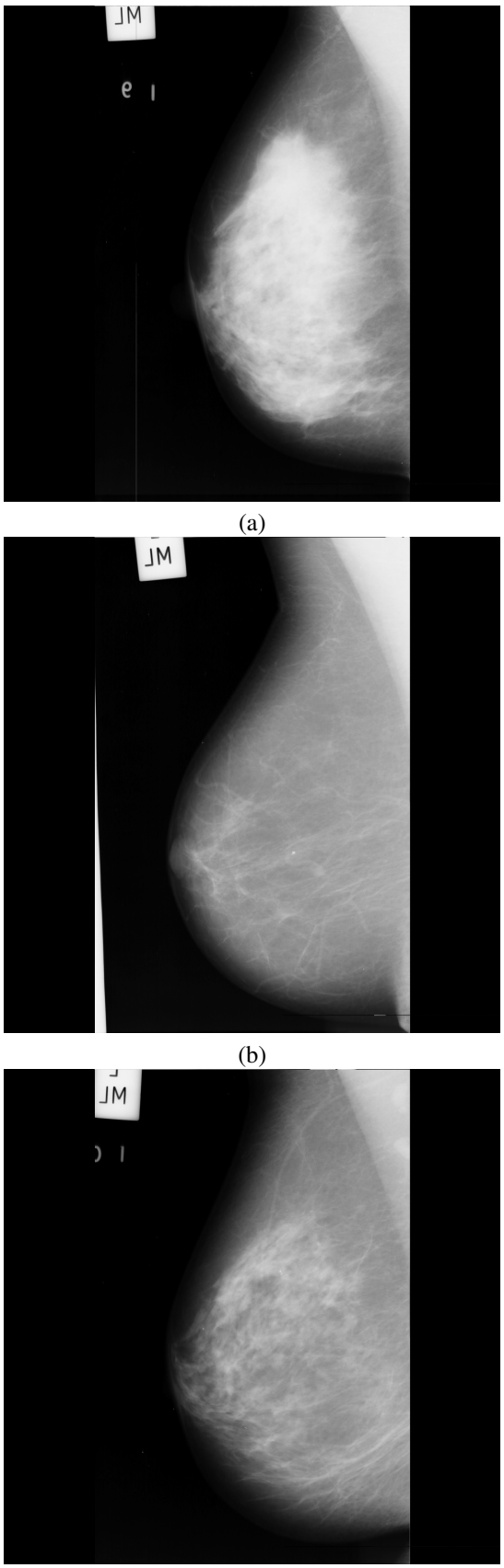
In this section, first we give information about the dataset, then describe the test methods and report the test results.

The dataset used in the experiments is a subset of the MIAS database. MIAS set consists of 322 mammographic images of size 1024x1024 pixels with detailed annotation. These images have one out of the three background tissue types which are fatty, fatty-glandular and dense-glandular, and some of them contain various types of benign or malignant abnormalities [11]. Since the problem in this study is just the tissue type classification and not abnormality detection/classification, the subset was constructed using only the images without any abnormalities. The used subset has 25 images in the class of dense-glandular (labeled with D), 37 images in the class of fatty (labeled with F), and 29 images in the class of fatty-glandular (labeled with G) tissues. Image samples from the MIAS database that belong to the dense-glandular, fatty and fatty-glandular classes are shown in Figure 5.1 (a), (b) and (c), respectively.

These images are cropped manually. The goal of this procedure is to deal only with the region of interest (ROI) and extract features from these regions. The resultant images are saved in an uncompressed image format in order to avoid the data loss arising from compression and format changes.

The cropped images are normalized before the feature extraction process. This is performed by finding the minimum and maximum intensity values in the image, subtracting the minimum from all pixels, multiplying all pixel values by 255 and dividing them to the difference (maximum - minimum).

SIFT features are extracted from all cropped and normalized images. Several keypoints are found in an image. The data for a class is constructed combining all the descriptors from the images which belong to that class. A SIFT descriptor consists of 128 integers



(a) (b) (c)

Figure 5.1: Image samples from the MIAS database which are (a) dense-glandular (b) fatty (c) fatty-glandular.

taking values between 0 and 255. SIFT implementation in a software library called OpenPR [18] was used for feature extraction.

In order to split the data into training and test parts, each class is clustered using the ISODATA algorithm. This algorithm started from a large number of clusters and finished with 13 clusters in each class by combining similar clusters. Half of the samples from each cluster are assigned to the training set. The objective here is to make the training set contain all types of information that exist in the data. This step helps minimizing the probability of having a training set which does not represent the data well. This procedure provides a bag-of-words representation of each class, where cluster means constitute the codebook of the class. Each image in a class is related to many code vectors in the codebook of its own class.

The extracted features are classified using support vector machines (SVM), Gaussian Mixture Models classifiers (GMM) and Learning Vector Quantization (LVQ) algorithms.

Support vector classification was performed using Weka with LIBSVM package. Weka, which stands for Waikato Environment for Knowledge Analysis, is a powerful machine learning and data mining tool which has been developed by University of Waikato in New Zealand under GNU General Public License. It has a plugin architecture and it is developed in Java [19]. LIBSVM is an integrated software for support vector classification [20]. It has interfaces for many programming languages and software and there is a wrapper for Weka called WLSVM [21].

For Learning vector quantization (LVQ), another Weka plugin called WEKA Classification Algorithms is used [17]. This plugin provides a collection of algorithms such as LVQ, self organizing maps and artificial immune system and it is licensed under GNU General Public License.

A Matlab toolbox, PRTools [22], is used for the GMM classification. This software package is developed by Delft Pattern Recognition Group in Delft University of Technology and is free for academic purposes.

Results are reported as the numbers of true and false classified observations and in terms of precision, recall and accuracy. These are statistical measures that are

calculated using the number of true positives (tp), the number of true negatives (tn), the number of false positives (fp) and the number of false negatives (fn). Precision, recall and accuracy are formulated as in (**5.1**), (**5.2**) and (**5.3**), respectively. Precision is used for learning the ratio of correct sample assignments over all assignments to a class. Recall is the ratio of the number of samples of a class that are labeled correctly over the total predictions of the samples in that class. The information that accuracy reports is the percentage of the total correct predictions over all predictions.

$$\text{Precision} = \frac{tp}{tp+fp} \tag{5.1}$$

$$\text{Recall} = \frac{tp}{tp+fn} \tag{5.2}$$

$$\text{Accuracy} = \frac{tp+tn}{tp+tn+fp+fn} \tag{5.3}$$

We have evaluated the classification performance in four test cases. First the binary classification performance is reported by 10-fold cross validation using GMM, SVM and LVQ classifiers separately. The aim in using 10-fold cross validation was to eliminate the effects of the outliers that exist in the data.

In the second test case, we have used separate training and test sets in order to evaluate the size of the test set and make sure that the training and test sets do not overlap. Again, these tests were performed using GMM, SVM and LVQ classifiers.

In the third case, we have evaluated the size of the training size in SVM and LVQ classifiers. We have enlarged the dataset by adding new samples and replicating them. These tests were performed using the 10-fold cross validation scheme.

In the fourth test case we have performed three-class classification tests using the SVM and LVQ classifiers on the enlarged training set in a 10-fold cross validation scheme.

Table 5.1 reports the 10-fold cross-validation results obtained by using the GMM classifier. In this test, each class contains 2620 samples and modeled with a mixture of eight Gaussian components. From Table 5.1 it can be seen that the performance of the GMM is not acceptable.

Per-class average values calculated from Table 5.1 can be seen in Table 5.2. It can be seen that the worst separable class is the fatty-glandular class, and the other two have similar performances.

Table 5.1: 10-fold GMM classification results.

	D	F	Precision	Recall	Accuracy(%)
D	1777	843	0.615	0.678	62.6527
F	1114	1506	0.641	0.575	
Weighted average			0.628	0.627	
	G	D	Precision	Recall	Accuracy(%)
G	1218	1402	0.487	0.465	48.7595
D	1283	1337	0.488	0.510	
Weighted average			0.488	0.488	
	G	F	Precision	Recall	Accuracy(%)
G	1419	1201	0.497	0.542	49.7137
F	1434	1186	0.497	0.453	
Weighted average			0.497	0.497	

Table 5.2: Per-class averages for 10-fold GMM classification.

	Precision	Recall	Accuracy (%)
D	0.552	0.594	55.7061
F	0.569	0.514	56.1832
G	0.492	0.504	49.2366

In Table 5.3 the results obtained using support vector machines in the 10-fold cross-validation scheme are given. In these tests, a polynomial kernel of degree 3 was used, and the cost value of the classifier was set to 100. A normalization was applied to the data before classification. In this table, there are 1318 samples in class D, 2839 samples in class F and 1663 samples from class G. From Table 5.3 it can be seen that SVM results are much better than GMM results. The best classification is between dense-glandular and fatty classes.

Table 5.3: 10-fold SVM classification results.

	D	F	Precision	Recall	Accuracy(%)
D	455	863	0.702	0.345	74.5971
F	193	2646	0.754	0.932	
Weighted average			0.738	0.746	
	G	D	Precision	Recall	Accuracy(%)
G	1394	269	0.603	0.838	60.2482
D	916	402	0.599	0.305	
Weighted average			0.602	0.602	
	G	F	Precision	Recall	Accuracy(%)
G	450	1213	0.625	0.271	67.0591
F	270	2569	0.679	0.905	
Weighted average			0.659	0.671	

Table 5.4 presents the per-class averages calculated from Table 5.3. In this table we can see that the fatty class can be separated best from the other classes.

Table 5.4: Per-class averages 10-fold SVM classification.

	Precision	Recall	Accuracy (%)
D	0.651	0.325	67.4227
F	0.717	0.919	70.8281
G	0.614	0.555	63.6537

Table 5.5 presents the performance achieved by the learning vector quantization algorithm. A normalization was performed before the LVQ process. The LVQ setting is the HierarchicalLvq algorithm that employs Olvq1 as the base algorithm. The submodel algorithm performs a multipass operation using Lvq2.1 algorithm for both passes. All algorithms have the same learning rate of 0.1, 20 codebook vectors, 5000 training iterations and all of them were initialized using K-Nearest neighbor algorithm. In this table, there are 1318 samples in class D, 2839 samples in class F and 1663 samples from class G. As can be seen from Table 5.5, LVQ results are lower than SVM results except when classifying between the fatty and dense-glandular classes where the difference is very small, and LVQ performs better than GMM.

Table 5.5: 10-fold LVQ classification results.

	D	F	Precision	Recall	Accuracy(%)
D	646	672	0.618	0.49	74.2122
F	400	2439	0.784	0.859	
Weighted average			0.731	0.742	
	G	D	Precision	Recall	Accuracy(%)
G	1110	553	0.61	0.667	57.5981
D	711	607	0.523	0.461	
Weighted average			0.571	0.576	
	G	F	Precision	Recall	Accuracy (%)
G	670	993	0.529	0.403	64.6824
F	597	2242	0.693	0.79	
Weighted average			0.632	0.647	

Per-class average values calculated from Table 5.5 can be seen in Table 5.6. Average results of the LVQ are similar to the average results of the SVM with a little lower values.

Table 5.6: Per-class averages for 10-fold LVQ classification.

	Precision	Recall	Accuracy (%)
D	0.571	0.476	65.9052
F	0.739	0.825	69.4473
G	0.570	0.535	61.1403

From the first test case, it can be seen that the best results are achieved using the SVM algorithm. According to these results, it can be said that when the size of the training set is small SVM outperforms other classifiers. In this test case the performance of the GMM is not acceptable. From the average values, the accuracy of the classification of the fatty class is the best when using any of the three classifiers, and the fatty-glandular class has the worst performance.

Results of the second test case are given from Table 5.7 to Table 5.12. In Table 5.7, test results using GMM classifier with separate training and test sets can be seen. Each class was modeled using 8 Gaussians. In this table, the training set consists of 1976 samples from each class, and the test set consists of 830 observations from class G, 1404 observations from class F and 659 observations from class D. From Table 5.7 we can see that the performance of the GMM increased except when classifying between the fatty and fatty-glandular classes when compared to the results in the first test case. Still these results are very low.

Table 5.7: GMM results using separate training and test sets.

	D	F	Precision	Recall	Accuracy(%)
D	493	166	0.504	0.748	68.444
F	485	919	0.847	0.655	
Weighted average			0.737	0.684	
	G	D	Precision	Recall	Accuracy(%)
G	411	419	0.585	0.495	52.317
D	291	368	0.468	0.558	
Weighted average			0.533	0.523	
	G	F	Precision	Recall	Accuracy(%)
G	381	449	0.350	0.459	48.1647
F	709	695	0.608	0.495	
Weighted average			0.512	0.482	

In Table 5.8 per-class averages calculated from Table 5.8 can be seen. Average values in Table 5.8 show the same improvement that can be observed from Table 5.7.

Table 5.8: Per-class averages for GMM classification using separate training and test sets.

	Precision	Recall	Accuracy (%)
D	0.486	0.653	60.3805
F	0.728	0.575	58.3044
G	0.468	0.477	50.2409

In Table 5.9, test results using SVM classifier with separate training and test sets can be seen. In this table, the training set consists of 1318, 2839 and 1663 samples from classes D, F and G, respectively, and the test set consists of 1311, 2835 and 1654 samples from classes D, F and G, respectively. It can be seen from Table 5.9 that the SVM classification results are similar to that given in the first test case.

Table 5.9: SVM results using separate training and test sets.

	D	F	Precision	Recall	Accuracy(%)
D	449	862	0.696	0.342	74.4814
F	196	2639	0.754	0.931	
Weighted average			0.736	0.745	
	G	D	Precision	Recall	Accuracy(%)
G	1404	250	0.602	0.849	60.2698
D	928	383	0.605	0.292	
Weighted average			0.603	0.603	
	G	F	Precision	Recall	Accuracy(%)
G	430	1224	0.621	0.26	66.8969
F	262	2573	0.678	0.908	
Weighted average			0.657	0.669	

Per-class average values calculated from Table 5.9 can be seen in Table 5.10. Average values of the SVM classification results are similar to that reported in the first test case.

Table 5.10: Per-class averages for SVM classification using separate training and test sets.

	Precision	Recall	Accuracy (%)
D	0.651	0.317	67.3756
F	0.716	0.920	70.6892
G	0.612	0.555	63.5834

In Table 5.11, test results using LVQ with separate training and test sets can be seen. In this table, the training set consists of 1318, 2839 and 1663 samples from classes D, F and G, respectively, and the test set consists of 1311, 2835 and 1654 samples from

classes D, F and G, respectively. Similar to SVM results, there is no significant change in LVQ results when compared to the first test case, as can be seen from Table 5.11.

Table 5.11: LVQ results using separate training and test sets.

	D	F	Precision	Recall	Accuracy(%)
D	655	656	0.606	0.5	73.9267
F	425	2410	0.786	0.85	
Weighted average			0.729	0.739	
	G	D	Precision	Recall	Accuracy(%)
G	1119	535	0.62	0.677	58.8196
D	686	625	0.539	0.477	
Weighted average			0.584	0.588	
	G	F	Precision	Recall	Accuracy(%)
G	584	1070	0.519	0.353	64.09
F	542	2293	0.682	0.809	
Weighted average			0.622	0.641	

Per-class averages calculated from Table 5.11 are given in Table 5.12. It can be seen that average values are consistent with the results given in Table 5.11, which means there is no significant change in the performance of LVQ in this test case.

Table 5.12: Per-class averages for LVQ classification using separate training and test sets.

	Precision	Recall	Accuracy (%)
D	0.573	0.489	66.3732
F	0.734	0.830	69.0083
G	0.570	0.515	61.4548

From the results of the second test case, we can say that SVM and LVQ results are consistent to the first test case since there is no significant difference between two cases. Although the performance of the GMM is improved, these results are still not acceptable. In the remaining tests we considered not using GMM since it has a high computational complexity which causes long test durations.

Selection of the optimal training set is an important issue in classification. In the third test case, we have used different training sets to examine whether the training sets yield overfitting. The training sets are enlarged by either replicating the data or including new data into the set. In order to see the effects of enlarging the data size, 10-fold cross-validation tests were performed on a dataset created by adding new

samples and using the same samples many times. The results obtained by the SVM and the LVQ are reported in Table 5.13 and Table 5.15, respectively. For the SVM, as before, a polynomial kernel of degree 3 was used. The cost value was set to 100 and a normalization was applied before classification. In Table 5.13 and Table 5.15, the dataset contains 31500 samples from each class. Since the number of samples increased, more codewords are needed thus we have used 5000 codewords.

Table 5.13: 10-fold SVM results using the enlarged dataset.

	D	F	Precision	Recall	Accuracy(%)
D	24603	6897	0.760	0.781	76.7111
F	7775	23725	0.775	0.753	
Weighted average			0.767	0.767	
	G	D	Precision	Recall	Accuracy(%)
G	20869	10631	0.663	0.663	66.3000
D	10600	20900	0.663	0.663	
Weighted average			0.618	0.618	
	G	F	Precision	Recall	Accuracy(%)
G	20506	10994	0.674	0.651	66.8095
F	9916	21584	0.663	0.685	
Weighted average			0.668	0.668	

From Table 5.13 we can see that there is no significant improvement except the results of the classification between the fatty-glandular and the dense-glandular classes.

Per-class average values calculated from Table 5.13 can be seen in Table 5.14. From this table we can see that the most separable class is the fatty class, and we can see the improvement in the fatty-glandular class.

Table 5.14: Per-class averages for 10-fold SVM classification using the enlarged dataset.

	Precision	Recall	Accuracy (%)
D	0.712	0.722	67.0056
F	0.719	0.719	71.7603
G	0.669	0.657	66.5548

It can be seen from Table 5.15 that the performance of the LVQ algorithm highly increased when using the enlarged dataset.

Per-class average values calculated from Table 5.15 can be seen in Table 5.16. This table shows that the average classification performances for each class is above 90%.

Table 5.15: 10-fold LVQ results using the enlarged dataset.

	D	F	Precision	Recall	Accuracy(%)
D	29609	1891	0.923	0.94	93.0683
F	2476	29024	0.939	0.921	
Weighted average			0.931	0.931	
	G	D	Precision	Recall	Accuracy(%)
G	29813	1687	0.954	0.946	95.0429
D	1436	30064	0.947	0.954	
Weighted average			0.950	0.950	
	G	F	Precision	Recall	Accuracy(%)
G	28761	2739	0.886	0.913	89.7587
F	3713	27787	0.910	0.882	
Weighted average					

Table 5.16: Per-class averages for 10-fold LVQ classification using the enlarged dataset.

	Precision	Recall	Accuracy (%)
D	0.935	0.947	94.0556
F	0.925	0.902	91.4135
G	0.920	0.930	92.4008

It is seen from Table 5.13 that enlarging the training set does not significantly change the performance of the SVM classifier. It is observed from Table 5.16 that increasing the number of training vectors significantly increases the performance of the LVQ algorithm. This is because the LVQ is a learning based scheme that can be affected from the inadequate number of training data.

As the fourth test case, 3-class classification tests were performed using the SVM and LVQ classifiers on the enlarged dataset. Same classifier parameters as in the third test case were used. Results using SVM and LVQ are given in Table 5.17 and Table 5.18, respectively.

Table 5.17: 3-class SVM classification results using 10-fold cross validation.

	D	F	G	Precision	Recall	Accuracy(%)
D	20064	5926	5510	0.574	0.637	54.6783
F	4623	21167	5710	0.559	0.672	
G	10264	10796	10440	0.482	0.331	
Weighted average				0.538	0.547	

From Table 5.17 we can see that the performance of SVM algorithm significantly decreased in the three-class classification scheme.

Table 5.18: 3-class LVQ classification results using 10-fold cross validation.

	D	F	G	Precision	Recall	Accuracy(%)
D	29142	1402	956	0.924	0.925	90.6667
F	1406	27890	2204	0.895	0.885	
G	974	1878	28648	0.901	0.909	
Weighted average				0.907	0.907	

In Table 5.18 we can see that that LVQ gives similar results to the two-class classification results. LVQ learns the data well with sufficient number of samples and enough number of codevectors to represent them. According to these results, dense-glandular density is the best classified one and fatty class is the worst one. However, the variance in the precision and recall values of different tissue types are not high.

6. CONCLUSION

In this study a system is proposed for breast tissue density classification in mammograms. This system may be considered as a preprocessing step in computer aided diagnosis systems working on mammograms and content based medical image retrieval systems.

Breast tissue density is highly related to the risk of cancer development. Furthermore, the masses in breast may hide in the dense tissue. Thus detection of a malignant mass becomes a hard task, which may hinder the early detection of cancer and even lead to death.

We have used a subset of the MIAS dataset which includes mammographic images from different tissue types and some of these images contain benign or malignant masses. There are three tissue density classes in this set which are fatty, fatty-glandular and dense-glandular. These categories have overlapping global and visual image characteristics. In other words, two images from the same class may have different image characteristics, and two images from different classes may have similar characteristics. This structure of the data led us to use local image features. For this purpose, we have used the scale-invariant feature transform (SIFT) and applied the bag-of-features method in order to achieve a better representation of the data and to select the training set optimally.

Using the features extracted from the labeled images, three different classifiers was designed which are the Gaussian mixture models classifier (GMM), support vector machines (SVM) and learning vector quantization (LVQ). Our objective in using different classifiers was to evaluate the performance of them and to find the best method that is suitable for our purpose.

In our experiments, the performance of the GMM was not acceptable with our settings. On enlarged datasets we did not test GMM since the duration of these tests got longer.

We have observed that LVQ is the most efficient method when the computation time is considered.

The results show that in smaller datasets, SVM outperforms other methods. It is observed that 10-fold cross validation results are coherent with the results achieved by using separate training and test sets.

When the dataset is enlarged by adding new samples or replicating the same vectors, the performance of learning vector quantization highly increases while only subtle improvements are observed in the performance of support vector machines.

The performance of the SVM algorithm decreased in the three-class classification tests when compared to the two-class classification tests.

Three class classification using LVQ algorithm gives over 90% accuracy when there are sufficient number of samples in the training set and enough codewords to model them. LVQ algorithm learns the data better when the size of the training set is large, even if the samples are replicated.

According to the results of the three-class classification using LVQ algorithm, the most separable class is dense-glandular and the least separable one is the fatty class. However, there is only little differences between the performances in the classification of different classes.

The developed system may be used in computer aided diagnosis systems since it contributes to the reliability and automaticity of the CAD systems. Similarly, it can be used as a part of the content based medical image retrieval systems because it may reduce the work load and time by reducing the dataset to be searched over while affecting the accuracy in a positive way.

As future work, we plan using different feature extraction and selection methods and different datasets including digital mammograms, and reporting our results in terms of the BI-RADS categories.

REFERENCES

[1] **Zhang, G.**, **Wang, W.**, **Moon, J.**, **Pack, J.K. and Jeon, S.I.**, (2011). A Review of Breast Tissue Classification in Mammograms, *Proceedings of the 2011 ACM Symposium on Research in Applied Computation (RACS)*, pp.232–237.

[2] **Oliver, A.**, **Freixenet, J. and Zwiggelaar, R.**, (2005). Automatic Classification of Breast Density, *Proceedings of the 2005 IEEE International Conference on Image Processing (ICIP)*, volume 2, pp.1258–61.

[3] **American College of Radiology**, (1998). *Illustrated Breast Imaging Reporting and Data System BIRADS*, American College of Radiology, 3rd edition.

[4] **Oliver, A.**, **Freixenet, J.**, **Marti, R.**, **Pont, J.**, **Perez, E.**, **Denton, E.R. and Zwiggelaar, R.**, (2008). A Novel Breast Tissue Density Classification Methodology, *IEEE Transactions on Information Technology in Biomedicine*, **12(1)**, 55–65.

[5] **Bosch, A.**, **Munoz, X.**, **Oliver, A. and Marti, J.**, (2006). Modeling and Classifying Breast Tissue Density in Mammograms, *Prodeecings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pp.1552–1558.

[6] **Ferrari, R.**, **Rangayyan, R.**, **Desautels, J.**, **Borges, R. and Frere, A.**, (2004). Automatic Identification of the Pectoral Muscle in Mammograms, *IEEE Transactions on Medical Imaging*, **23(2)**, 232–245.

[7] **Oliver, A.**, **Freixenet, J.**, **Marti, R. and Zwiggelaar, R.**, (2006). A Comparison of Breast Tissue Classification Techniques, *Proceedings of the MICCAI 2006*, pp.872–879.

[8] **de Oliveira, J.E.E.**, **de Albuquerque Araújo, A. and Deserno, T.M.**, (2011). Content-Based Image Retrieval Applied to BI-RADS Tissue Classification in Screening Mammography, *World Journal of Radiology*, **3(1)**, 24–31.

[9] **Petroudi, S. and Brady, M.**, (2011). Breast Density Characterization Using Texton Distributions, *Proceedings of the 2011 IEEE Annual International Conference of theEngineering in Medicine and Biology Society (EMBC)*, pp.5004–5007.

[10] **Wang, J.**, **Li, Y.**, **Zhang, Y.**, **Xie, H. and Wang, C.**, (2011). Bag-of-Features Based Classification of Breast Parenchymal Tissue in the Mammogram via Jointly Selecting and Weighting Visual Words, *Proceedings of the*

2011 Sixth International Conference on Image and Graphics (ICIG), pp.622–627.

[11] **Suckling, J.**, **Parker, J.**, **Dance, D.**, **Astley, S.**, **Hutt, I.**, **Boggis, C.**, **Ricketts, I.**, **Stamatakis, E.**, **Cerneaz, N.**, **Kok, S.**, **Taylor, P.**, **Betal, D. and Savage, J.**, (1994). The Mammographic Images Analysis Society Digital Mammogram Database, *Experta Medica International Congress Series*, **1069**, 375–378.

[12] **Lowe, D.G.**, (2004). Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, **60(2)**, 91–110.

[13] **Csurka, G.**, **Dance, C.R.**, **Fan, L.**, **Willamowski, J. and Bray, C.**, (2004). Visual Categorization with Bags of Keypoints, *Proceedings of the International Workshop on Statistical Learning in Computer Vision (ECCV)*, pp.1–22.

[14] **Webb, A.R.**, (2002). *Statistical Pattern Recognition*, John Wiley & Sons, 2nd edition.

[15] **Kohonen, T.**, (2001). *Self-Organizing Maps*, Springer-Verlag, 3rd edition.

[16] **Reynolds, D.A.**, (2008). Gaussian Mixture Models, *Encyclopedia of Biometric Recognition*, 659–663.

[17] **Algorithms, W.C.**, `http://wekaclassalgos.sourceforge.org/`, date retrieved: 3rd April 2012.

[18] **OpenPR**, `http://www.openpr.org.cn/`, date retrieved: 15th February 2012.

[19] **Hall, M.**, **Frank, E.**, **Holmes, G.**, **Pfahringer, B.**, **Reutemann, P. and Witten, I.H.**, (2009). The WEKA Data Mining Software: An Update, *SIGKDD Explorations*, **11(1)**.

[20] **Chang, C.C. and Lin, C.J.**, (2011). LIBSVM: A Library for Support Vector Machines, *ACM Transactions on Intelligent Systems and Technology*, **2(3)**.

[21] **EL-Manzalawy, Y. and Honavar, V.**, (2005). *WLSVM: Integrating LibSVM into Weka Environment*.

[22] **van der Heijden, F.**, **Duin, R.P.**, **de Ridder, D. and Tax, D.M.**, (2004). *Classification, Parameter Estimation and State Estimation - An Engineering Approach Using Matlab*, John Wiley & Sons.

CURRICULUM VITAE

Name Surname: Sezer Kutluk

Place and Date of Birth: Ilgın/Konya, 21st February 1985

Address: Istanbul Technical University
Department of Electronics and Communications Engineering
Multimedia Signal Processing and Pattern Recognition Laboratory
34469 Maslak Istanbul Turkey

E-Mail: sezer.kutluk@gmail.com

B.Sc.: Electrical-Electronics Engineering Department, Istanbul University

List of Publications:

- Çırakman Ö., **Kutluk S.**, Günsel B., Çalıkuş O., 2012: Mobil Ortamda Ürün Algılama Amaçlı Bir Sözlük Ağacı Gerçeklemesi - A Vocabulary-Tree Implementation For Mobile Product Recognition. *IEEE 20th Signal Processing and Communications Applications Conference (SİU)*, 2012 Fethiye, Turkey.

- **Kutluk S.**, Günsel B., 2010: ITU MSPR TRECVID 2010 Video Copy Detection System. *TRECVID 2010*, 2010 Gaithersburg, MD, USA.

- Gürsoy O., **Kutluk S.**, Günsel B., Şengör N., 2010: Negatif Olmayan Matris Ayrıştırma ile İkili Video Kıyımlama - Binary Video Hashing by Non-Negative Matrix Factorization, *IEEE 18th Signal Processing and Communications Applications Conference (SİU)*, 2010 Diyarbakır, Turkey.