

EARLY PROGNOSIS OF BREAST  
CANCER USING IMAGE PROCESSING  
AND MACHINE LEARNING

A THESIS  
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND  
COMPUTER ENGINEERING  
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE  
OF ABDULLAH GUL UNIVERSITY  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
MASTER

By  
Sena Būşra Yengeç Taşdemir  
December 2018

Sena Būşra  
Yengeç Taşdemir

EARLY PROGNOSIS OF BREAST CANCER USING IMAGE  
PROCESSING AND MACHINE LEARNING

AGU  
2018

EARLY PROGNOSIS OF BREAST CANCER  
USING IMAGE PROCESSING AND MACHINE  
LEARNING

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER  
ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE OF  
ABDULLAH GUL UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER

By

Sena Büşra YENGEÇ TAŞDEMİR

December 2018

## SCIENTIFIC ETHICS COMPLIANCE

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

Sena Būşra YENGEÇ TAŞDEMİR



## REGULATORY COMPLIANCE

M.Sc. thesis titled “EARLY PROGNOSIS OF BREAST CANCER USING IMAGE PROCESSING AND MACHINE LEARNING” has been prepared in accordance with the Thesis Writing Guidelines of the Abdullah Gül University, Graduate School of Engineering & Science.

Prepared By

Sena Büşra YENGEÇ TAŞDEMİR

Advisor

Dr. Zafer AYDIN

Head of the Electrical and Computer Engineering Program

Prof. Dr. Vehbi Çağrı GÜNGÖR

## ACCEPTANCE AND APPROVAL

M.Sc. thesis “Early Prognosis Of Breast Cancer Using Image Processing And Machine Learning” prepared by Sena Büşra Yengeç Taşdemir has been accepted by the jury in the Electrical and Computer Engineering Graduate Program at Abdullah Gül University, Graduate School of Engineering & Science.

26/12/2018

### JURY:

Dr. Zafer AYDIN :.....  
Dr. Bekir Hakan AKSEBZECI :.....  
Doç. Dr. Alper BAŞTÜRK :.....

### APPROVAL:

The acceptance of this M.Sc. thesis has been approved by the decision of the Abdullah Gül University, Graduate School of Engineering & Science, Executive Board dated ...../...../ ..... and numbered .....

...../...../ .....

**(Date)**

Graduate School Dean  
Prof. Dr. İrfan ALAN

## ABSTRACT

# EARLY PROGNOSIS OF BREAST CANCER USING IMAGE PROCESSING AND MACHINE LEARNING

Sena Büşra YENGEÇ TAŞDEMİR  
M.Sc. in Electrical and Computer Engineering Department

**Supervisor:** Dr. Zafer AYDIN

December-2018

Among females, leading cause of cancer death and the most common cancer type is breast cancer. Early detection is vital because it reduces the mortality rate. Digital mammography is a widespread medical imaging technique that is used for early detection and diagnosis of the breast cancer. Automatic detection of tumorous area from the digital mammography image helps to locate the abnormal tissues, which may be analyzed further by a radiologist. It has two main stages: feature extraction and classification. In this work, numerous feature extraction methods have been tested such as 2D-DWT, HOG, Haralick's textural features, TAS, LBP, Zernike and GLCM. In order to select the most suitable classifier, the following classifiers also have been tested: random forest, logistic regression, k-nearest neighbors, naïve Bayes, decision tree, support vector machines, Adaboost, radial basis function network, multilayer perceptron, convolutional neural network. Based on comprehensive experiments, the optimum combination of feature extraction, feature selection and classification methods are identified. The proposed method, which employs CLAHE as image pre-processing tool, 2D-DWT, HOG, Haralick as feature extraction methods, wrapper as the feature selection method and random forest as the classifier, attained an accuracy of 87.5%.

*Keywords: Breast Cancer, ROI detection, Haralick Features, Wavelet Decomposition, HOG Features, Random Forest Classifier*

**ÖZET**

**GÖRÜNTÜ İŞLEME VE MAKİNE ÖĞRENMESİ**  
**YÖNTEMİYLE ERKEN MEME KANSERİ TEŞHİSİ**

Sena Büşra YENGEÇ TAŞDEMİR  
Elektrik ve Bilgisayar Mühendisliği Bölümü Yüksek Lisans

**Tez Yönetisi:** Dr. Zafer AYDIN

Aralık-2018

Kadınlarda, kanser ölümünün önde gelen nedeni ve en sık görülen kanser türü meme kanseridir. Erken teşhisi ölüm oranını azaltır, bu nedenle erken teşhis çok önemlidir. Dijital mamografi, meme kanserinin erken teşhisi ve tanısında kullanılan yaygın bir tıbbi görüntüleme tekniğidir. İlgili bölgenin (ROI) otomatik olarak saptanması, bir radyolog tarafından daha fazla analiz edilebilecek şekilde anormal alanları işaretlenmesine yardımcı olur. ROI'nin otomatik algılanması, özellik çıkarımı ve sınıflandırılması olmak üzere iki ana aşamaya sahiptir. Öznitelik çıkarma, görüntüyü bir bilgisayar için daha anlaşılır olan başka bir boyuta dönüştürür. İkinci adım, sınıflandırıcı tarafından yapılan kararı (normal veya ROI) içerir. Bu çalışmada, 2D-DWT, HOG, Haralick'in dokusal özellikleri, TAS, LBP, Zernike ve GLCM gibi farklı öznitelik çıkarma yöntemleri kullanılmıştır. Sistemin performansını değerlendirmek için, gerçekleştirilen sınıflandırıcılar; rastgele orman, lojistik regresyon, k-en yakın komşular (k-NN), naîve Bayes, karar ağacı, destek vektör makinesi (SVM), Adaboost, radyal temelli fonksiyon ağı (RBF-NN), çok katmanlı algılayıcı (MLP), konvolüsyonel sinir ağı (CNN) kullanılmıştır. Kapsamlı deneyler neticesinde, optimum başarıyı veren özellik çıkarma, özellik seçimi ve sınıflandırma yöntemleri tespit edilmiştir. Önerilen yeni ROI tanıma yönteminde görüntü ön işleme aracı olarak CLAHE, öznitelik çıkarmak için 2D-DWT, HOG, Haralick, özellik seçim yöntemi olarak wrapper ve sınıflandırıcı olarak rastgele orman yöntemi kullanılmış ve % 87.5'lik bir doğruluk oranı elde edilmiştir.

*Anahtar kelimeler: Meme Kanseri, İlgili Bölge tespiti, Haralick'in Dokusal Öznitelikleri, Dalgacık Ayırımı, HOG Öznitelikleri, Rastgele Orman Sınıflandırıcısı*

# Acknowledgements

I would like to thank my advisor Dr. Zafer AYDIN for his invaluable guide, help and support.

I would like to thank my wonderful family members, my father İbrahim YENGEÇ and my mother F. Reyhan YENGEÇ for their wholehearted support and love.

I would especially like to thank my better half, my husband, Kasım TAŞDEMİR for his love, support, encouragement, tolerance and assistance.

I would like to thank my friend Zeynep ŞENEL ÜNAL for her support and encouragement.





# Table of Contents

|   |    |
|---|----|
| 1. Introduction .....   | 1  |
| 1.1 Breast Anatomy .....  | 2  |
| 1.2 Mammographic abnormality types .....  | 3  |
| 1.2.1 Circumscribed lesions .....   | 3  |
| 1.2.2 Speculated masses .....   | 4  |
| 1.3 CAD System .....  | 5  |
| 1.4 Literature review for Brest Cancer Detection .....  | 5  |
| 2. Methods .....  | 9  |
| 2.1 Datasets, image enhancement and feature extraction methods for breast mass detection .....    | 9  |
| 2.1.1 Datasets .....  | 9  |
| 2.1.2 Image enhancement .....   | 11 |
| 2.1.2.1 Histogram Equalization .....  | 11 |
| 2.1.2.2 Dilation .....  | 12 |
| 2.1.2.3 Erosion .....   | 12 |
| 2.1.2.4 Opening .....   | 12 |
| 2.1.2.5 Closing .....   | 12 |
| 2.1.2.6 White top-hat transform .....   | 12 |
| 2.1.2.7 Black top-hat transform .....   | 12 |
| 2.1.3 Feature extraction methods .....  | 13 |
| 2.1.3.1 Multi-resolution analysis using two-dimensional discrete wavelet transform (2D-DWT) ..... | 13 |
| 2.1.3.2 Histogram of oriented gradients (HOG) .....   | 14 |
| 2.1.3.3 Haralick's Textural Features .....  | 15 |
| 2.1.3.4 Threshold adjacency statistics (TAS) .....  | 16 |
| 2.1.3.5 Local Binary Patterns (LBP) .....   | 17 |
| 2.1.3.6 Zernike moments .....   | 18 |
| 2.1.3.7 Gray-level co-occurrence matrix .....   | 18 |
| 2.1.4 Feature selection .....   | 19 |
| 2.1.4.1 Correlation feature selection (CFS) .....   | 19 |
| 2.1.4.2 Correlation attribute evaluation .....  | 19 |
| 2.1.4.3 Gain ratio attribute evaluation (GR) .....  | 20 |
| 2.1.4.4 Information gain attribute evaluation .....   | 20 |
| 2.1.4.5 Principal component analysis (PCA) .....  | 20 |
| 2.1.4.6 Relief-based feature selection algorithm .....  | 20 |
| 2.1.4.7 Symmetrical uncertainty (SU) .....  | 21 |
| 2.1.4.8 Wrapper .....   | 21 |
| 2.2 Classification Methods .....  | 22 |
| 2.2.1 Decision tree learning .....  | 22 |

|  |    |
|--|----|
| 2.2.2 Random forest .....                                    | 23 |
| 2.2.3 K-nearest neighbors .....                              | 24 |
| 2.2.4 Naïve bayes classifier .....                           | 24 |
| 2.2.5 Logistic regression .....                              | 25 |
| 2.2.6 Support vector machines (SVM) .....                    | 25 |
| 2.2.7 Adaptive boosting (AdaBoost) .....                     | 25 |
| 2.2.8 Artificial Neural Network (ANN) .....                  | 26 |
| 2.2.9 Multilayer Perceptron (MLP) .....                      | 27 |
| 2.2.10 Radial basis function network (RBFN) .....            | 28 |
| 2.2.11 Convolution neural network (CNN) .....                | 29 |
| 2.3 Cross validation techniques (CV) .....                   | 30 |
| 2.3.1 Ten-fold cross validation (10-fold CV) .....           | 30 |
| 2.3.2 Leave-one instance out cross validation (LOO-CV) ..... | 30 |
| 2.4 Classifier evaluation metrics .....                      | 30 |
| 2.4.1 Confusion Matrix .....                                 | 30 |
| 2.4.2 Accuracy .....   | 31 |
| 2.4.3 Precision .....  | 31 |
| 2.4.4 Recall (Sensitivity) .....                             | 31 |
| 2.4.5 Specificity .....                                      | 32 |
| 2.4.6 Area under curve score (AUC) .....                     | 32 |
| 2.4.7 F1 score .....   | 32 |
| 2.4.8 False positive rate (FPR) .....                        | 32 |
| 2.5 Proposed Method .....                                    | 33 |
| 3. Experiments and Analysis .....                            | 35 |
| 3.1 Evaluation of feature extraction methods .....           | 35 |
| 3.2 Image enhancement on mammographic images .....           | 73 |
| 4. Conclusions and Future Prospects .....                    | 87 |
| 4.1 Conclusions .....  | 87 |
| 4.2 Future Prospects .....                                   | 88 |
| BIBLIOGRPHY .....  | 89 |

# List of Figures

|   |    |
|---|----|
| Figure 1.1 Various type of cancer occurrences among females according to HDI index of countries .....   | 1  |
| Figure 1.1.1. Structure of the female breast.....   | 2  |
| Figure 1.1.2 Small structures that produces the milk.....   | 3  |
| Figure 1.2.1.1 Circumscribed lesions is marked with red circle (from MIAS database) .....   | 3  |
| Figure 1.2.1.2 Speculated mass is marked with red circle (from MIAS database) .....   | 4  |
| Figure 1.2.1.2 Microcalcification is marked with red circle (from MIAS database) .....  | 5  |
| Figure 2.1.1.1 A ROI marked as a red circle in a mammogram image (from Pilot dataset). White circles represent mole markers .....   | 10 |
| Figure 2.1.2.1.1 CLAHE Operation.....   | 11 |
| Figure 2.1.2.2.1 Histogram Equalization Process .....   | 12 |
| Figure 2.1.3.1.1 Sub-bands of 2 level wavelet decomposition .....   | 13 |
| Figure 2.1.3.1.2 One level wavelet decomposition of mammographic ROI.....   | 13 |
| Figure 2.1.3.2.1 Histogram of Oriented Gradients visualization of abnormal ROI .....  | 15 |
| Figure 2.1.3.3.1 Spatial Information of GLCM .....  | 16 |
| Figure 2.1.3.3.2 Computation of Matrices with unit distance and four directions. (a) Gray Level Values of the Input Image (b) 0° GLCM c) 45° GLCM d) 90° GLCM e) 135° GLCM..... | 16 |
| Figure 2.1.3.4.1 Process of counting the number of white pixels .....   | 17 |
| Figure 2.1.3.5.1 Local Binary Pattern of a matrix element .....   | 17 |
| Figure 2.1.3.5.2 Calculation of the score using a binary array .....  | 18 |
| Figure 2.1.4.8.1 Diagram of Wrapper feature selection .....   | 21 |
| Figure 2.2.1.1 Simple decision tree .....   | 23 |
| Figure 2.2.2.1 Random forest .....  | 23 |
| Figure 2.2.3.1 K-nearest neighbors.....   | 24 |
| Figure 2.2.5.1 Logistic Function.....   | 25 |
| Figure 2.2.7.1 Visualization of hyperplane .....  | 26 |
| Figure 2.2.8.1 Visualization of a ANN .....   | 27 |
| Figure 2.2.9.1 One hidden layer multilayer perceptron .....   | 28 |
| Figure 2.2.10.1 Construction of RBFN.....   | 28 |
| Figure 2.2.10.2 Gaussian radial function with different $\beta$ values .....  | 29 |
| Figure 2.2.11.1 CNN layers .....  | 29 |
| Figure 2.2.2.1.1 Confusion Matrix.....  | 31 |
| Figure 2.2.2.6.1 Visualization of a ROC curve.....  | 32 |
| Figure 2.3.1 Flow Diagram of the Proposed Algorithm .....   | 33 |
| Figure 3.1.1 ROC curves of Random Forest and AdaBoost .....   | 57 |
| Figure 3.1.2 AdaBoost occurrence of the accuracy versus number of iterations for 50 iterations.....   | 66 |
| Figure 3.1.3 Random Forest occurrence of the accuracy versus number of trees for 25 trees.....  | 66 |
| Figure 3.1.4 SVM occurrence of the accuracy versus different parameter for $C = 2^{13}$ and $\gamma = 2^{-5}$ .....   | 67 |
| Figure 3.1.5 The mammogram (a) is traversed window by window and labeled using the model, labels reconstruct the mammogram (b).....   | 68 |
| Figure 3.1.6 ROC curves of the Haralick and the proposed Wavelet-Haralick-HOG descriptor (feature selection is applied in LOOCV setting for each method).....                   | 72 |

|  |    |
|--|----|
| Figure 3.2.1 a is a sample from MIAS dataset and b is from pilot dataset.....            | 73 |
| Figure 3.2.2 a) original mammogram and B) CLAHE processed version of original .....      | 74 |
| Figure 3.2.3 a) original image b) histogram equalized version of the original image..... | 74 |
| Figure 3.2.4 Original mammogram and Dilation applied mammogram .....                     | 75 |
| Figure 3.2.5 Original mammogram and Erosion applied mammogram.....                       | 75 |
| Figure 3.2.6 Original mammogram and Opening applied mammogram .....                      | 76 |
| Figure 3.2.7 Original mammogram and Closing applied mammogram.....                       | 76 |
| Figure 3.2.8 Original mammogram and Top-Hat (white top-hat) applied mammogram.....       | 77 |
| Figure 3.2.9 Original mammogram and Black Top-Hat applied mammogram .....                | 77 |



# List of Tables

|   |    |
|---|----|
| Table 3.1.1 Haralick features are extracted from Pilot dataset and evaluated using 10-fold cross validation with random forest classifier without normalization .....             | 36 |
| Table 3.1.2 Haralick features are extracted from the dataset which is constructed from 31 abnormal and 62 normal instances and evaluated using 10-fold CV with random forest..... | 37 |
| Table 3.1.3 Haralick features are extracted from the dataset which is constructed from 31 abnormal and 93 normal instances and evaluated using 10-fold CV with random forest..... | 37 |
| Table 3.1.4 Haralick features are extracted from the dataset which contains 31 abnormal and 124 normal instances and evaluated using 10-fold CV with random forest.....           | 38 |
| Table 3.1.5 Haralick features are extracted from the dataset which contains 31 abnormal and 124 normal instances and evaluated using 10-fold CV with random forest.....           | 38 |
| Table 3.1.6 Haralick features are extracted from balanced dataset, normalized and evaluated using 10-fold cross validation with random forest classifier.....                     | 39 |
| Table 3.1.7 Haralick features are extracted from unbalanced dataset, normalized and evaluated using 10-fold cross validation with random forest classifier .....                  | 40 |
| Table 3.1.8 Haralick features are extracted from balanced dataset evaluated using 10-fold cross validation using AdaBoost classifier .....  | 41 |
| Table 3.1.9. Haralick features are extracted from imbalanced dataset evaluated using 10-fold cross validation using AdaBoost classifier .....                                     | 41 |
| Table 3.1.10 Haralick features are extracted from balanced dataset evaluated using 10-fold cross validation using KNN classifier.....   | 42 |
| Table 3.1.11 Haralick features are extracted from imbalanced dataset evaluated using 10-fold cross validation using KNN classifier.....   | 42 |
| Table 3.1.12 Haralick features are extracted from balanced dataset evaluated using 10-fold cross validation using Logistic Regression classifier.....                             | 43 |
| Table 3.1.13. Haralick features are extracted from unbalanced dataset evaluated using 10-fold cross validation using Logistic Regression classifier.....                          | 43 |
| Table 3.1.14 Haralick features are extracted from balanced dataset evaluated using 10-fold cross validation using Naive Bayes classifier .....                                    | 44 |
| Table 3.1.15 Haralick features are extracted from unbalanced dataset evaluated using 10-fold cross validation using Naive Bayes classifier .....                                  | 44 |
| Table 3.1.16 Haralick features are extracted from balanced dataset evaluated using 10-fold cross validation using Decision Tree classifier.....                                   | 45 |
| Table 3.1.17 Haralick features are extracted from imbalanced dataset evaluated using 10-fold cross validation using Decision Tree classifier .....                                | 45 |
| Table 3.1.18 Haralick features are extracted from balanced dataset evaluated using 10-fold cross validation using SVM classifier.....   | 46 |
| Table 3.1.19 Haralick features are extracted from unbalanced dataset evaluated using 10-fold cross validation using SVM classifier.....   | 46 |
| Table 3.1.20. Haralick features are extracted from balanced dataset evaluated using 10-fold cross validation using RBF-NN classifier.....   | 47 |
| Table 3.1.21 Haralick features are extracted from unbalanced dataset evaluated using 10-fold cross validation using RBF-NN classifier.....  | 47 |
| Table 3.1.22 LBP features are extracted from balanced dataset, evaluated using 10-fold cross validation with random forest classifier.....  | 48 |

|   |    |
|---|----|
| Table 3.1.23 LBP features are extracted from balanced dataset, normalized and evaluated using 10-fold cross validation with random forest classifier.....   | 48 |
| Table 3.1.24 LBP features are extracted from unbalanced dataset, evaluated using 10-fold cross validation random forest classifier.....   | 49 |
| Table 3.1.25 LBP features are extracted from unbalanced dataset, normalized and evaluated using 10-fold cross validation random forest classifier.....  | 49 |
| Table 3.1.26 LBP features are extracted from balanced dataset and evaluated using 10-fold cross validation AdaBoost classifier .....  | 50 |
| Table 3.1.27. LBP features are extracted from imbalanced dataset and evaluated using 10-fold cross validation AdaBoost classifier .....   | 50 |
| Table 3.1.28 LBP features are extracted from balanced and imbalanced dataset, evaluated using 10-fold cross validation with different classifiers.....  | 51 |
| Table 3.1.29 LBP features are extracted from balanced dataset, evaluated using 10-fold cross validation with Logistic Regression classifier .....   | 52 |
| Table 3.1.30 TAS features are extracted from balanced and imbalanced dataset, evaluated using 10-fold cross validation with different classifiers.....  | 52 |
| Table 3.1.31 TAS features are extracted from balanced dataset, evaluated using 10-fold cross validation with Naive Bayes classifier.....  | 53 |
| Table 3.1.32 Zernike features are extracted from balanced and imbalanced dataset, evaluated using 10-fold cross validation with different classifiers .....   | 53 |
| Table 3.1.33 Zernike features are extracted from balanced dataset, evaluated using 10-fold cross validation with K-NN classifier .....  | 54 |
| Table 3.1.34 GLCM matrix statistical features are extracted from balanced and imbalanced dataset, evaluated using 10-fold cross validation with different classifiers .....   | 55 |
| Table 3.1.35 GLCM matrix statistical features are extracted from balanced dataset, evaluated using 10-fold cross validation with RBF Network classifier .....   | 55 |
| Table 3.1.36 HOG features are extracted from balanced and imbalanced dataset, evaluated using 10-fold cross validation with different classifiers.....  | 56 |
| Table 3.1.37 HOG features are extracted from balanced dataset, evaluated using 10-fold cross validation with Naive Bayes classifier.....  | 57 |
| Table 3.1.38 Summarization of different classifiers and feature descriptors f-measure for class 1 .....   | 58 |
| Table 3.1.39 Feature selection methods evaluated using different classifiers .....  | 62 |
| Table 3.1.40 Accuracy metrics of KNN when k=20.....   | 63 |
| Table 3.1.41 Accuracy metrics of Random Forest when number of trees equal to 25.....  | 63 |
| Table 3.1.42 Accuracy metrics of RBF Network when number of clusters equal to 15 .....  | 64 |
| Table 3.1.43 Accuracy metrics of AdaBoost when number of iterations equal to 50.....  | 65 |
| Table 3.1.44 Accuracy metrics of SVM when $C = 2^{13}$ and $\gamma = 2^{-5}$ .....  | 65 |
| Table 3.1.45 LOO-CV performed on best classifiers using Haralick features.....  | 67 |
| Table 3.1.46 Different combination of classifiers has been performed using Logistic and SVM meta learner .....  | 69 |
| Table 3.1.47 Different combination of features are extracted from balanced dataset, feature selection is made by Wrapper method, then selected and non-selected versions is evaluated using 10-fold cross validation with Random Forest classifier using default parameters ..... | 70 |
| Table 3.1.48 LOO-CV performed on different feature extraction algorithms using Random forest algorithm.....   | 71 |
| Table 3.1.49 LOO-CV performed on proposed feature extraction algorithm using different classifiers....  | 72 |
| Table 3.2.1. Proposed methods features are extracted from MIAS database and evaluated using Random Forest classifier with 10-fold CV.....   | 78 |

|   |    |
|---|----|
| Table 3.2.2 Proposed methods features are extracted from CLAHE applied MIAS database and evaluated using Random Forest classifier with 10-fold CV ..... | 78 |
| Table 3.2.3 MLP accuracy result of one-fold and ten-folds on normalized dataset.....  | 79 |
| Table 3.2.4 MLP accuracy result of one-fold and ten-folds on dataset .....  | 79 |
| Table 3.2.5 MLP accuracy results and optimum parameters for different optimization techniques .....   | 81 |
| Table 3.2.6 MLP accuracy results and optimum parameters for 2 and 2 hidden layer MLP .....  | 82 |
| Table 3.2.7 MLP accuracy results of previous experiments .....  | 83 |
| Table 3.2.8 Accuracy values of Machine Learning Methods that have been performed on MIAS dataset  | 84 |
| Table 3.2.9 Accuracy metrics of feature selection methods that have been performed on MIAS dataset using Random Forest classifier .....                 | 85 |
| Table 3.2.10 Classifiers have been evaluate using LOO-CV.....   | 85 |





*To my beloved family..*

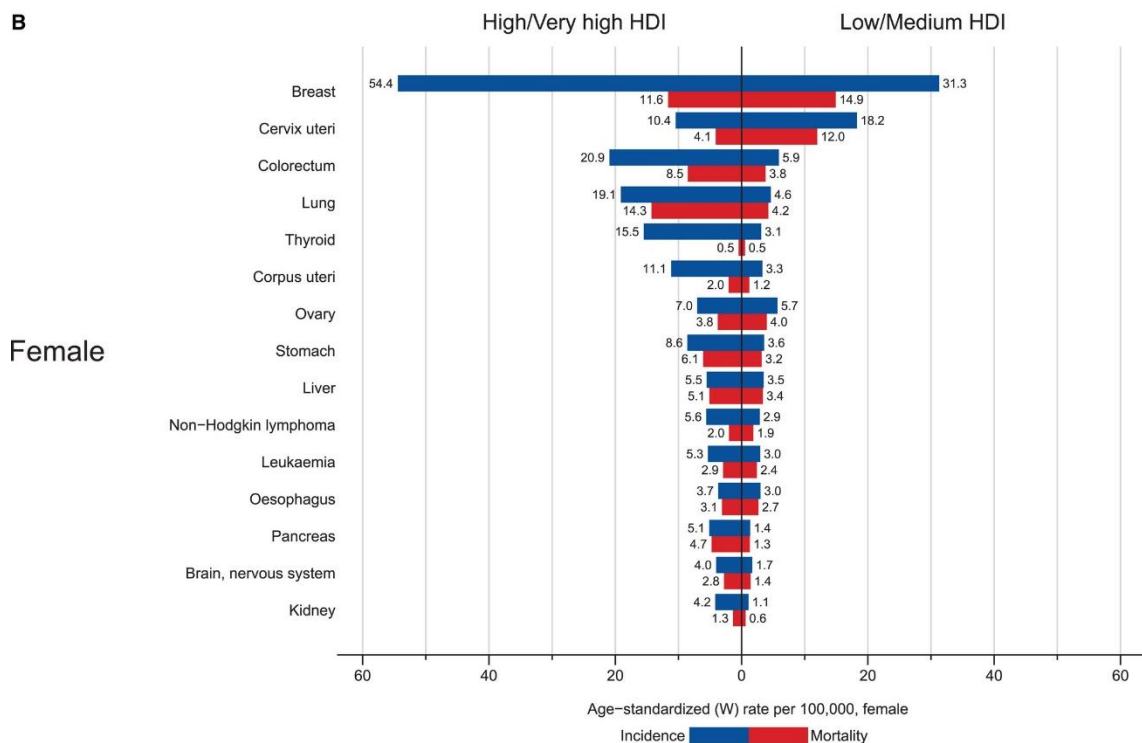


# Chapter 1

## Introduction

Every living creature is constructed from cells. All cells have a certain job to do and their life cycle contains reproducing new cells. The cells produce new cells by dividing itself in a decent way. Old or damaged cells die, the new cells take their places. Cancer occurs when the cells starts to uncontrolled cell proliferation which later forms tumor [1]. Not all cancer types have tumors moreover, tumors can be benign or malignant. Benign tumors are not cancerous while, malignant tumors are cancerous growth [2].

Cancer can start any part of body. If it starts at the breast, this type of cancer is called as breast cancer. Figure 1.1 describes the cancer type occurrences among woman according to human development index (HDI) of countries. Among females, leading cause of cancer death and most common cancer type is breast cancer (Figure 1.1) [3].



**Figure 1.1** Various type of cancer occurrences among females according to HDI index of countries [3].

Early detection of breast cancer is very important in order to develop an efficient treatment methodology. There are several techniques to examine the breast; MRI,

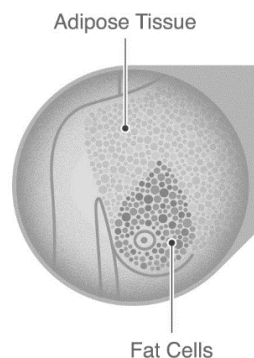
ultrasound and mammographic examination. Mammography often outdo other alternatives in any aspect. It is cheaper, easy, not time consuming [4].

However, the signs of the cancer are very subtle at the early stages. Therefore, expert radiologists can misdiagnose an important proportion of the cases from mammography [5]. To reduce this error rate, computer aided detection (CAD) systems have been developed.

A CAD system typically consists of three main stages: pre-processing, feature extraction and classification. Choosing the right method for each of these steps are important for the accuracy of a CAD system [6]. Furthermore, to facilitate the decision process, detecting regions of interest (ROI) automatically can help to localize the cancer tissue better instead of extracting features directly from the original high-resolution mammography image. This thesis is based on designing a novel CAD system using image processing and machine learning tools.

## 1.1 Breast anatomy

Female breast is mostly constructed from fat cells which is called adipose tissue (Figure 1.1.1) [7]. The primary function of the female breast is to produce milk which can be produced by lobes. Lobes contain smaller lobules in its structures. Figure 1.1.2 shows lobes, lobules and milk ducts that connects lobules and lobes [8]. Breast cancer usually starts to grow at lobes and lobules [7].



**Figure 1.1.1. Structure of the female breast [7]**

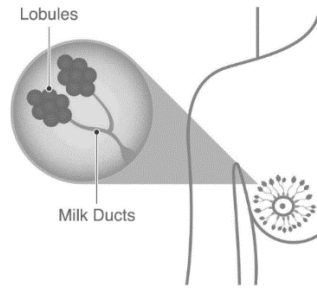


Figure 1.1.2 Small structures that produces the milk [7]

## 1.2 Mammographic abnormality types

Any region that does not look normal in mammogram is suspicious. The abnormalities in mammogram are circumscribed lesions, microcalcifications and speculated regions [4].

### 1.2.1 Circumscribed lesions

A focused white area on mammogram can be perceived as tumors. There are two type of tumors; benign and malignant. Benign tumor means non-cancerous. While malignant tumors are cancerous.

If tumors contour is very discriminative it is called as circumscribed lesions [9]. Figure 1.2.1.1 shows a mammogram instance, lesion is marked with red circle. Circumscribed lesions are usually accepted as benign.



Figure 1.2.1.1 Circumscribed lesions is marked with red circle (from MIAS database)

## 1.2.2 Speculated masses

A tartarated tissue with irregular shape forms the speculated mass [9]. Figure 1.2.1.2 shows an instance of speculated mass which is a well-known sign of malignancy.

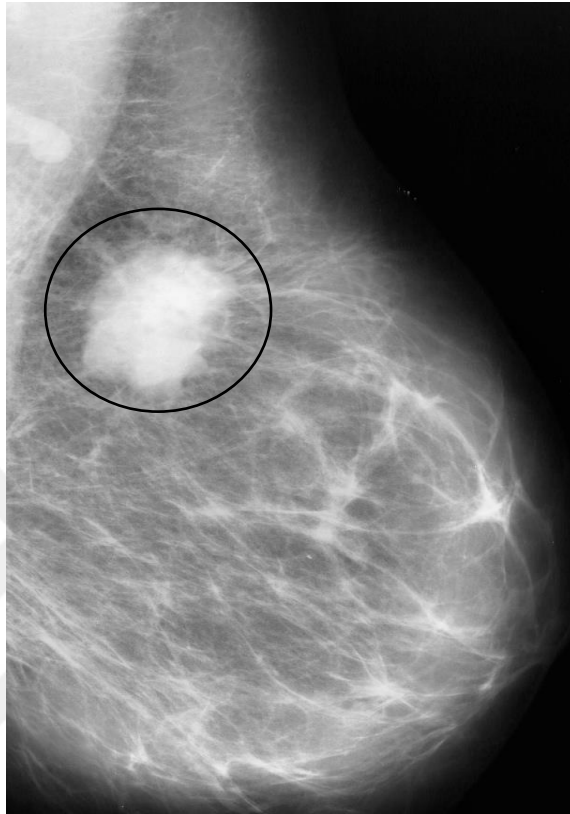


Figure 1.2.1.2 Speculated mass is marked with red circle (from MIAS database)

## 1.2.3 Microcalcifications

Calcium deposits can be received as sign of a cancer which can be seen as bright point on mammogram. If the calcium deposits are small, it is called microcalcification [10]. Figure 1.2.1.3 shows an instance of microcalcification. Microcalcifications are usually accepted as benign.



Figure 1.2.1.2 Microcalcification is marked with red circle (from MIAS database)

### 1.3 CAD System

CAD uses computer systems for early detection of abnormalities in the mammogram. Early detection is very important to reduce the mortality rate [5]. CAD system is constructed from preprocessing, feature extraction and classification.

Preprocessing stage enhances the visibility of cancer affected area from normal breast tissue. Feature extraction part converts the information of the image into another domain which is more understandable for a computer. Classification is the decision-making stage where computer decides the mammogram is either benign or malignant using machine learning techniques [6].

### 1.4 Literature review for Breast Cancer Detection

Various combination of feature extraction and machine learning methods have been used in order to develop a CAD system for breast cancer detection. Ericeria et al. [11] proposed the variogram function as a texture feature descriptor and used SVM as a classifier on a dataset that contains 150 abnormal, 30 normal images. They obtained accuracy of 90.26% and specificity of 85.37%.

Reyad et al. [12] used LBP and some statistical measure as feature extraction algorithm. The evaluation of the algorithm is performed using SVM classifier on DDSM database. The best accuracy measure is achieved when both LBP and statistical measures are used.

Tahmasbi et al. [13] extracted ROIs manually from mammogram and used them as input of the system. Zernike moments and multi-layer perceptron are combined and evaluated on MIAS database. The overall accuracy and AUC score are 96.43%, 97.6% respectively.

Junior et al. [14] proposed a method that distinguishes the tissue as mass or not-mass on DDSM databases mammograms. They also used manually cropped ROIs as input of the system. They used Geary's Coefficient and Moran's Index to texture characterization and SVM as classifier. The method achieves 98.13% of sensitivity and 98.36% specificity.

Ergin et al. [15] used Dense Scale Invariant Feature Transform (DSIFT), Histogram of Oriented Gradient (HOG), Local Configuration Pattern (LCP) as feature extraction algorithm and used SVM, k-NN, Decision Tree and Fisher's LDA as classifier on whole mammogram images of DDSM dataset. 10-fold cross validation method have been employed and best accuracy measure of 99.80 was achieved by using HOG features and SVM classifier.

Beura et al. [16] extracted ROI from the mammogram image and extracted features from the ROIs. The feature extraction part includes two level, first they applied discrete wavelet transform to ROI and computed the GLCM of sub-wavelet transformed images. In order to eliminate redundant features, they used f-test and t-test. Back propagation neural network (BPNN) is used as classifier and the evaluation of algorithm is done on both MIAS and DDSM database. For MIAS and DDSM database the accuracy measures are 98% and 98.8% respectively.

Rouhi et al. [17] extracted ROIs using segmentation, later segmented tumor regions was used to extract features. To segment the regions, they used trained artificial neural network. As features they used intensity, shape and textural features which is later eliminated by genetic search algorithm to select relevant features. Multilayer perceptron was adopted as classifier on both DDSM and MIAS database. They obtained accuracies for MIAS and DDSM are 90.16, 96.47 respectively.

Gedik [18] proposed a method that extracts features using fast finite shearlet transform. Extracted features are eliminated using t-test. SVM used as classifier on both

MIAS and DDSM database. The accuracy results 97%, 100% for MIAS and DDSM achieved using 5-fold CV.

Khan et al. [19] used Gabor features that extracted from ROIs. In order to eliminate irrelevant features, they adopted particle swarm optimization (PSO). SVM was used as classifier on DDSM dataset. The accuracy of the algorithm is 98.2%.

Shastri et al. [20] implemented a combination of HOG algorithm and Gabor filter which is called Histogram of Texture (HOT). The features are extracted from ROIs. They tested the algorithm on IRMA database using 2-fold CV. The algorithm achieves the accuracy of 95%.

Pak et al. [21] determined the ROIs automatically and improved the quality of images using super resolution and non-subsampled contourlet transform. The textural features are extracted. Adaboost classifier have been used on MIAS database, and overall accuracy is 91.43%.

Jasmine et al. [22] used wavelet transform feature extraction on MIAS dataset. Artificial neural network has been used to detect abnormalities of the tissue. True detection rate is 87%.

Mutaz et al. [23] segmented ROIs. They extracted textural features using GLCM from ROI. GLCM features was extracted from four different angles and two-pixel distance. The classification has been done by using ANN on DDSM database. They achieved the accuracy of 91.67%.

Jona et al. [24] used mini-MIAS dataset. They extracted textural features from GLCM matrix. They evaluated the relevant features using two different optimization technique which are genetic algorithm and particle swarm optimization. They adopted SVM to eliminate the algorithms performance. They obtained accuracy of 94%.

Görgel et al. [25] extracted features using wavelet coefficients. They used a private database which contains 66 mammogram images. Among 66 images 28 images were abnormal and 38 images were normal. The classification part is done by using SVM classifier. They obtained accuracy of 84.8%.

Berbar [26] extracted both contourlet (CT) features and GLCM texture features. The contourlet features are represented as CT1 and CT2. CT1 applies contourlet at level 4, while CT2 applies contourlet at levels [4 3 2 1]. The experiment is one on DDSM dataset using SVM classifier using different parameters. Best accuracy measure of 98.69% is achieved.

Bovis and Singh [27] used Haralick's textural features on MIAS database. They constructed five GLCM and extracted Haralick's features. They tested the performance using ANN and 10-fold CV. The average accuracy of 77% was obtained.





# Chapter 2

## Methods

### 2.1 Datasets, image enhancement and feature extraction methods for breast mass detection

#### 2.1.1 Datasets

The datasets were acquired from both mini-MIAS (Mammographic Image Analysis Society) database and pilot dataset of the Digital Mammography Dream Challenge which was held on 2017.

The pilot set contains 500 images, 34 of them are labelled as abnormal and 466 as normal. The speculated region of interest of 31 abnormal mammographic images were marked by Assoc. Prof. Fahrettin Kılıç, who is an expert radiologist. The region of interests of the abnormal images were cropped and labelled manually which selects a rectangular region around the tumorous region. A similar cropping method was applied to 31 randomly selected normal mammograms, as well as, the size of the ROI was randomly selected. As a result, first ROI dataset has been constructed which contains 31 cancer positive and 31 cancer negative images. The size of the ROIs is approximately 73x68.



Figure 2.1.1.1 A ROI marked as a red circle in a mammogram image (from Pilot dataset). White circles represent mole markers

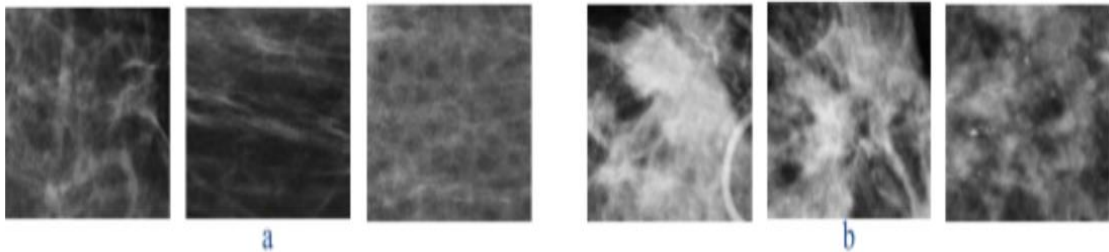


Figure 2.1.1.2 Mammogram ROI dataset: (a) Three samples from non-ROI images labeled as negative (b) Three samples from ROI images labeled as positive

The MIAS dataset consist of 322 digital mammograms, 109 abnormal and 213 normal images with the size of 1024x1024. The ROIs of the dataset were marked initially, the same ROI dataset construction process has been applied to MIAS. Although the image sizes of the first and the second ROI datasets differ for each image, the datasets can be used for feature extraction without any change.

Since the raw images of the MIAS-ROI set later will be used in Convolutional

Neural Network and Multi-Layer Perceptron, the size of the images must be same. For this reason, a third ROI dataset was produced with a fixed image size of 28x28 is average value of the ROIs in MIAS.

## 2.1.2 Image enhancement

The MIAS image quality is not good as Pilot dataset for this reason some image enhancement techniques have been applied.

### 2.1.2.1 Contrast limited adaptive histogram equalization (CLAHE)

CLAHE method is used for enhancing the contrast adaptively. Basis of operation is Adaptive Histogram Equalization (AHE). The image is divided into small blocks which is called tiles and AHE equalizes the histogram for each tile. If the tile has some noise, AHE amplifies the noise. Therefore, contrast limiting is used to avoid this problem. If there is any histogram bin which is above a contrast limit, those pixels are pruned and distributed uniformly, this process is shown in Figure 2.1.2.1.1 [28] [29]. After equalization, to correct the artificiality of the tile borders, linear interpolation is applied.

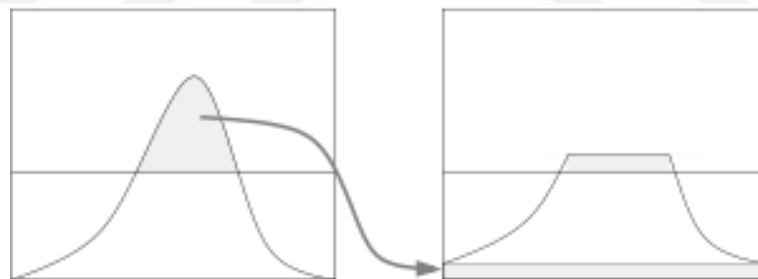
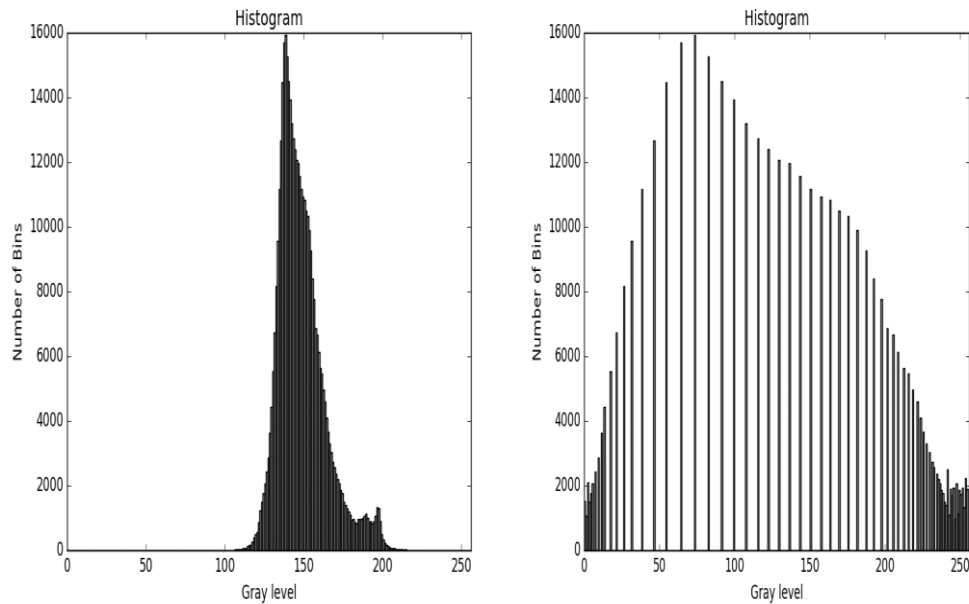


Figure 2.1.2.1.1 Clahe Operation [30]

### 2.1.2.2 Histogram Equalization

Histogram equalization is used to adjust the global contrast [31]. After this process, the intensities can be more even as shown in Figure 2.1.2.2.1.



**Figure 2.1.2.2.1 Histogram Equalization Process**

### **2.1.2.3 Dilation**

Dilation adds pixels to boundary of an object and fills holes using a structuring element [31].

### **2.1.2.4 Erosion**

This operation removes the boundary of the object using a structuring element [31].

### **2.1.2.5 Opening**

The morphological opening is an erosion followed by dilation [31].

### **2.1.2.6 Closing**

The morphological closing is a dilation followed by erosion. Closing removes small holes from foreground [31].

### **2.1.2.7 White top-hat transform**

The white top-hat transform calculates the difference between the input image and its opening [32].

### **2.1.2.8 Black top-hat transform**

The black top-hat transform calculates the difference of the input image and its closing [32].

## 2.1.3 Feature extraction methods

Feature extraction aims to capture the visual information content of an image by reducing the pixel data into another domain in order to facilitate the decision-making process.

### 2.1.3.1 Multi-resolution analysis using two-dimensional discrete wavelet transform (2D-DWT)

The discrete wavelet transform can be implemented as a combination of a down sampler and a filter bank that decomposes a 1D or a 2D signal (i.e. image) into sub-bands at different resolution levels [33]. The digital filter bank contains high-pass and low-pass filters. Because image pairs contain the high and low frequency information, the Wavelet transform can extract useful texture features from mammographic ROIs. The two-level Wavelet decomposition of a two-dimensional signal is shown in Figure 2.1.3.1.1.

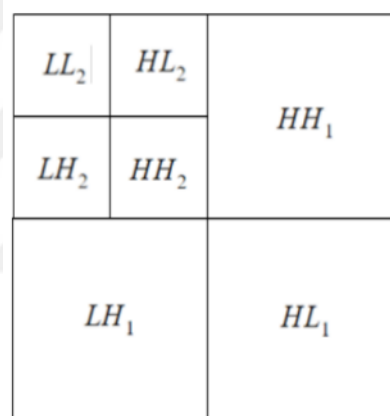


Figure 2.1.3.1.1 Sub-bands of 2 level wavelet decomposition

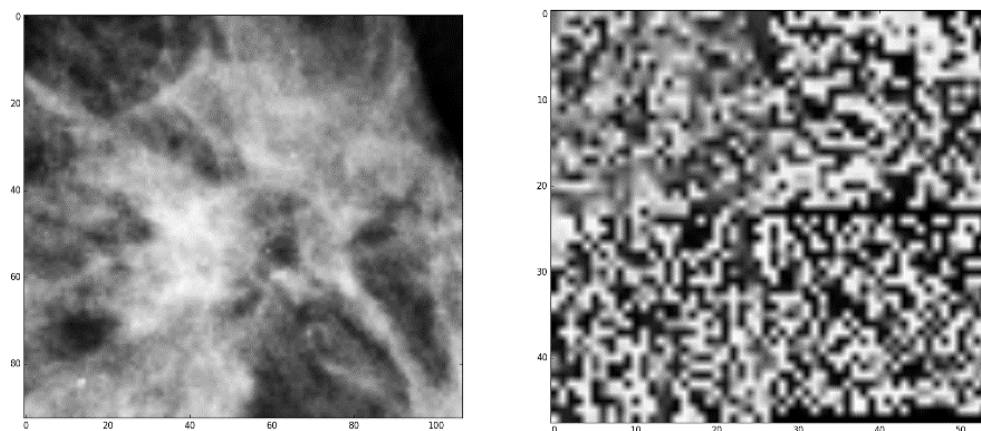


Figure 2.1.3.1.2 One level wavelet decomposition of mammographic ROI

To compute two-dimensional discrete wavelet, transform (DWT), one dimensional DWT is applied on each column and the row of the input image separately. Figure 2.1.3.1.2 illustrates the 2D-DWT of a given mammographic ROI. The kernel of the wavelet was Haar function. The right image in the figure illustrates the decomposition of the original image on the left into four sub-bands in frequency domain, which represents low-high (LH), high-low (HL), low-low (LL), high-high (HH) components corresponding to horizontal, vertical, approximation and diagonal respectively. Three of the sub-regions, HH, HL and LH contains the detail information for different orientation and resolution of the ROIs while the LL involves the coarse approximation at a particular resolution level [34]. The LL sub-band can be decomposed further into sub-bands repeatedly to increase the resolution of the Wavelet.

### 2.1.3.2 Histogram of oriented gradients (HOG)

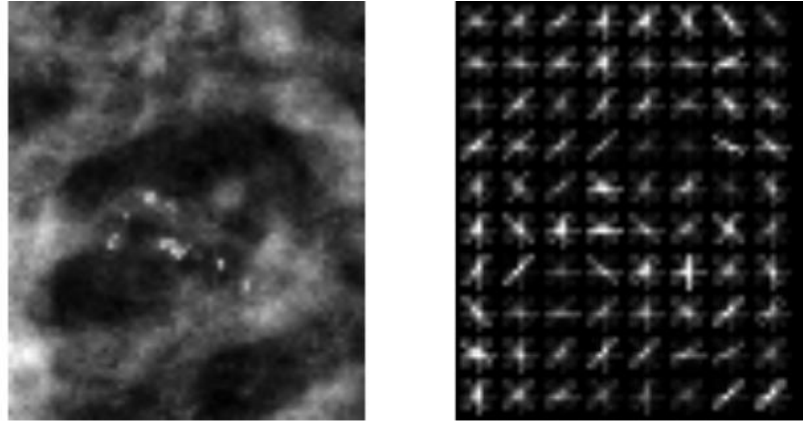
HOG features are used by the computer vision community to detect objects and localize them. It is based on the theory that a mass or shape can be distinguished by differential intensity histogram of the local intensity gradients or edge directions. Each mammographic ROI is divided into non-overlapping uniform windows. For each window, the differentials for each orientation are calculated. These are called the gradients  $G_x$  and  $G_y$  in x and y directions. The gradient which is related to differential is constructed from group of cells [35].

The second step includes calculating the magnitude and orientation of the gradient for each pixel in the image, which can be achieved using equation 2.1.3.2.1 and equation 2.1.3.2.1.

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (\text{Equation 2.1.3.2.1})$$

$$\theta(x, y) = \tan^{-1} \left( \frac{G_y}{G_x} \right) \quad (\text{Equation 2.1.3.2.2})$$

Figure 2.1.3.2.1 shows the visualization of a malignant ROI and its HOG representation. The HOG feature extraction method extracts 16 features and it is summarized in **Algorithm 2.1.3.2.1**.



**Figure 2.1.3.2.1 Histogram of Oriented Gradients visualization of abnormal ROI**

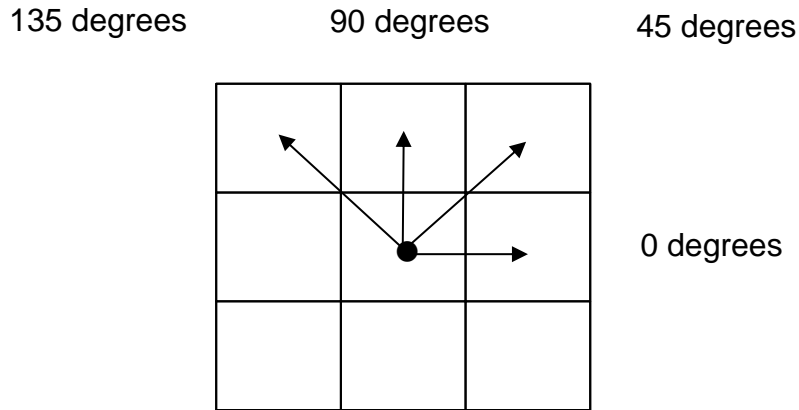
---

**Alorihm 2.1.3.2.1. HOG feature extraction**

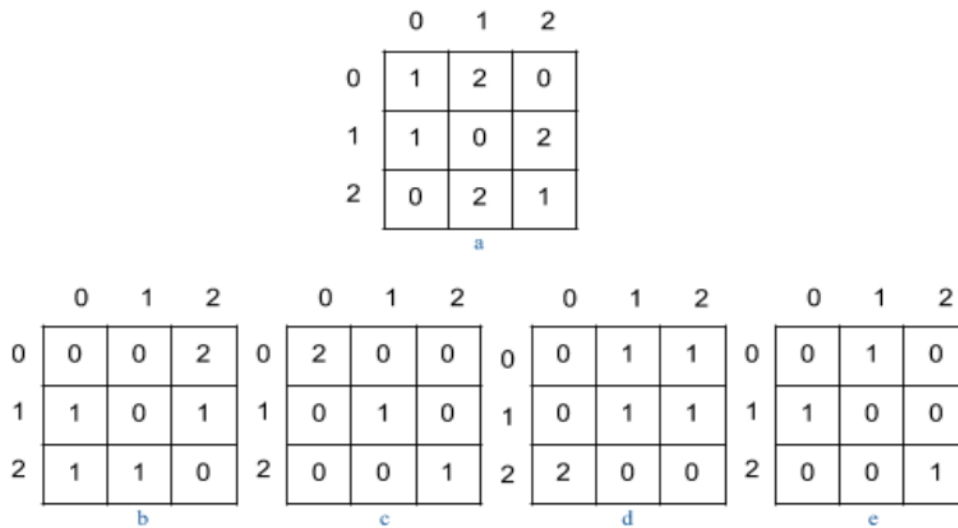
- 1 //input I: The mammographic ROI image
  - 2 //output: 16 HOG features
  - 3  $G_x$  = Sobel filter of input image (vertical);
  - 4  $G_y$  = Sobel filter of input image (horizontal);
  - 5 Magnitude, Angle = Calculate the magnitude and angle of  $G_x$  and  $G_y$ ;
  - 6 Number of Bins = 16;
  - 8 Bin =  $(\text{Number of Bins} * \text{Angle}) / 360$ ;
  - 9 Features = Count number of occurrences of each value in array of Bin using Magnitude as weight where minimum length is set equal to the number of bins;
  - 10 Return Features;
- 

**2.1.3.3 Haralick's Textural Features**

The texture of an image bears many characteristics that can be used to identify a ROI. In order to extract fourteen Haralick's features the gray level co-occurrence matrices (GLCM) are used [36]. GLCM is a technique that can be used to compute the texture information by capturing the spatial relationships of the pixels which is shown in Figure 2.1.3.3.1. The GLCM calculates the spatial relationship by counting the pixel pairs that have the same values for a given distance and direction. For this purpose, a co-occurrence matrix is computed for each of the four directions ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  and  $135^\circ$ ) with one-pixel distance. The size of the GLCM is equal to gray level value of the given image. The GLCM calculates the spatial relationship by counting the pixel pairs with constant values that occurs in a given distance and direction. This process is shown in Figure 2.1.3.3.2. 14 features for four directions can be derived from GLCM.



**Figure 2.1.3.3.1 Spatial Information of GLCM [37]**



**Figure 2.1.3.3.2 Computation of Matrices with unit distance and four directions. (a) Gray Level Values of the Input Image (b) 0° GLCM (c) 45° GLCM (d) 90° GLCM (e) 135° GLCM [38]**

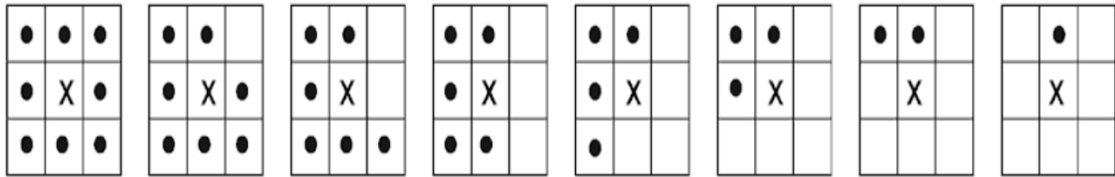
The gray level of the given image is three which is same size of the matrices. The co-occurrence matrices for each direction are calculated by counting the pixel pairs for the specified direction. After counting process numbers have been placed in/at an appropriate position. The fourteen Haralick features constructed from the following statistics contrast, entropy, homogeneity, correlation, mean of x-axis, mean of y-axis, energy, standard deviation, moment 1, moment 2, moment 3, distribution of the probability, ASM, mean, angular second moment (ASM), entropy [36].

### 2.1.3.4 Threshold adjacency statistics (TAS)

Threshold adjacency statistics are constructed as follows. First of all, original image is processed with three different threshold values, this process gives three binary



images. Threshold values are mean intensity value ( $\mu$ ), maximum of the mean ( $\mu+30$ ) and minimum of the mean value ( $\mu-30$ ), the minimum intensity value should be at least 30. Secondly, the statistics are found to reveal the dissimilarities between threshold images. The number of white pixels that has no white pixel neighbor is counted, the number of white pixels that has one neighbor white pixel is counted, this process repeated till the number of the white neighbor pixel reaches to eight. The process which is shown in the Figure 2.1.3.4.1, constructs the statistics [39].

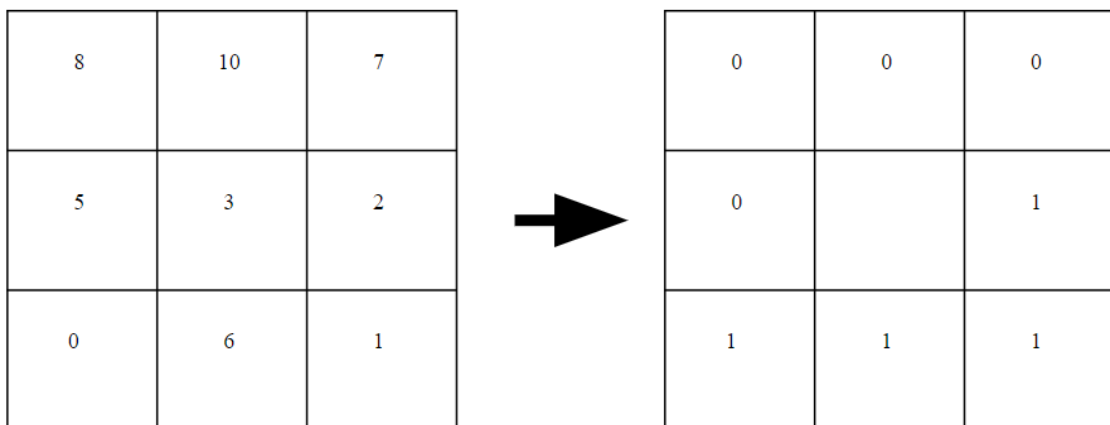


**Figure 2.1.3.4.1 Process of counting the number of white pixels [39]**

Each statistic is divided by the total number of white pixels to make normalization. This process is repeated for all three images and finally it gives 27 statistics.

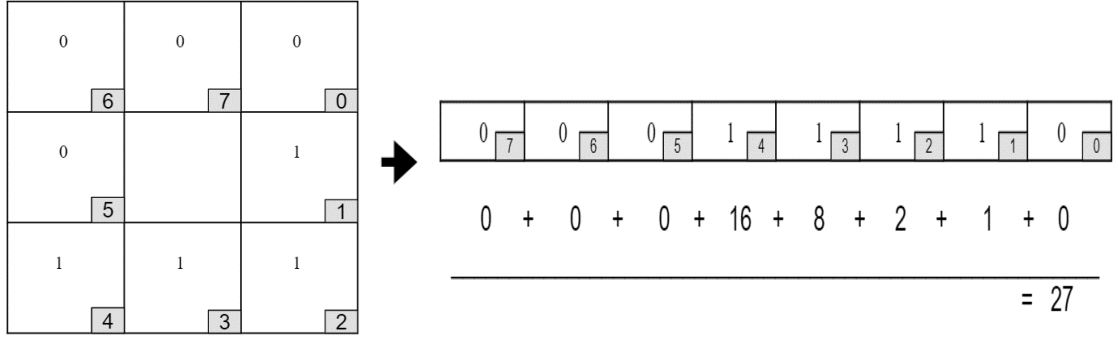
### 2.1.3.5 Local Binary Patterns (LBP)

Local Binary Pattern extracts features to classifying texture [40]. The method uses texture spectrum model. To construct the LBP feature vector, the image is traversed window by window. Windows are divided into cells and for each pixel in the cell is compared with its eight neighbors. If pixels intensity is greater than the neighbors' value, it is written 0 or else 1, it is shown in the Figure 2.1.3.5.1. Since it has the information of the 3x3 neighbors, there is  $2^8 = 256$  possible patterns [41]



**Figure 2.1.3.5.1 Local Binary Pattern of a matrix element [41].**

The results of this test are stored in a binary array which is later converted to decimal which is illustrated in the Figure 2.1.3.5.2.



**Figure 2.1.3.5.2 Calculation of the score using a binary array [41]**

The process is repeated for each pixel in the image. Finally, the 256-bin histogram of LBP feature vector is created.

### 2.1.3.6 Zernike moments

Zernike moments are not texture feature, it is a global measure to show the distribution of mass [42]. Moments are statistical measure which captures both global and geometric information of the entire image instead of a single boundary point. In order to extract the features using Zernike Moments first the image is converted into binary scaled numerical image. Then pixel coordinates are mapped to unit circle. Origin of the unit circle is center of the image. The outside of the unit circle is discarded. The unit circle image map is then converted into polar coordinates. Zernike is constructed from polynomials which is represented by  $V_{nm}(x, y)$  [42]. The polynomials are calculated using Equation 2.1.3.6.1.

$$V_{nm}(x, y) = V_{nm}(\rho, \theta) = R_{nm}(\rho)e^{jm\theta} \quad \text{(Equation 2.1.3.6.1)}$$

The  $\rho$  represents the length from origin to  $f(x, y)$  pixel and  $\theta$  represents the angle between the vector  $\rho$  and x-axis.  $R_{nm}(\rho)$  is radial polynomial which is calculated using the Equation 2.1.3.6.2.

$$R_{nm}(\rho) = \sum_{s=0}^{n-|m|/2} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2}-s\right)! \left(\frac{n-|m|}{2}-s\right)!} \rho^{n-2s} \quad \text{(Equation 2.1.3.6.2)}$$

The Zernike moments of order  $n$  with repetitions are calculated using the equation Equation 2.1.3.6.1 and Equation 2.1.3.6.3.

$$A_{nm} = \frac{n+1}{\pi} \sum_x \sum_y f(x, y) V_{nm}^*(\rho, \theta) dx dy \quad \text{(Equation 2.1.3.6.3)}$$

### **2.1.3.7 Gray-level co-occurrence matrix**

GLCM examines the spatial information the pixels in a subregion of an image. Creation of the GLCM matrix is described in the section 2.1.3.3 Haralick's Textural Features.

## **2.1.4 Feature selection**

The extracted features are mathematical descriptors, which can be employed by a classifier to decide. If the number of features is high, the classification model can suffer from the presence of irrelevant features as well as noise contained in features [43]. Further-more, as the number of dimensions increases the classification model can be prone to over-fitting, which reduces the prediction accuracy on unseen test examples. To address these problems, feature selection can be used. The major task is to select a feature set, which contains a sub-set of the original features that are relevant for predicting the output class.

### **2.1.4.1 Correlation feature selection (CFS)**

The CFS algorithm builds different sub-feature sets and evaluates the correlation between feature and class for each feature, according to CFS the correlation between feature and class should be robust but correlation between features should be weak. The CFS measures the merit of a sub-feature set to evaluate the relevant features [44]. On the other hand, the CFS in the Weka software uses three different measures which are relief, minimum description length and symmetrical uncertainty instead of merit.

### **2.1.4.2 Correlation attribute evaluation**

The correlation attribute evaluation works slightly different than the CFS. It builds feature subsets and it evaluates the features relevancy by measuring the Pearson's correlation coefficient for feature and the class [45]. However, The CFS calculates not only the correlation of feature and class, but also the correlation between all features.

### **2.1.4.3 Gain ratio attribute evaluation (GR)**

This feature selection method calculates the gain ratio according to class for each feature. Gain ratio measure is used for sorting the attributes. The algorithm chooses the feature that maximizes the GR to be on the top [44]. Process is repeated to sort the features

by relevance. The gain ratio is measured using the entropy measure. The entropy (H) is calculated for measuring the systems unpredictability.

$$Entropy(Class) = -\sum_{c \in class} P(c) \log_2(P(c)) \quad \text{(Equation 2.1.4.3.1)}$$

$$GainRatio(Class, Feature) = \frac{H(Class) - H(Class| Feature)}{H(feature)} \quad \text{(Equation 2.1.4.3.2)}$$

#### 2.1.4.4 Information gain attribute evaluation

The method calculates the information gain according to class for each feature [46]. The same process of gain ratio attribute evaluation is repeated.

$$InformationGain (Class, Feature) = H(Class) - H(Class| Feature) \quad \text{(Equation 2.1.4.4.1)}$$

#### 2.1.4.5 Principal component analysis (PCA)

The basic idea of the PCA is finding patterns in the data and transfer data into new feature space [43]. In order to find the patterns PCA uses eigen vectors (principal components) and eigen values of a covariance matrix. PCA projects high dimensional data into lower dimensions and retains most of the information. The eigen vectors determine the direction of the new feature space while the eigen values determine the magnitude.

#### 2.1.4.6 Relief-based feature selection algorithm

The algorithm calculates a score for each feature. Then, according to score the features are sorted and features whose rank is above a threshold will be selected [47]. To calculate the score, a score vector (w) is initialized. The size of the w is equal the number of the features (k) in the dataset. One feature vector (f) which belongs to a random instance is taken and compared with similar feature vectors which belongs to same and another class. The most similar same-class instance is near-hit, and the most similar different-class instance is near miss. For each feature the score is updated using the Equation 2.1.4.6.1. This process is repeated m times, note that the size of the data set is equal to n.

$$w_j = w_j - (f_j - Near_{hit})^2 + (f_j - Neat_{miss})^2 \quad \text{(Equation 2.1.4.6.1)}$$

### 2.1.4.7 Symmetrical uncertainty (SU)

This feature selection method calculates the symmetrical uncertainty according to class for each feature. SU measure is used for sorting the attributes. The algorithm chooses the feature that has the highest value of SU to be on the top [48]. Process is repeated to sort the features by relevance. SU is calculated using Mutual Information (MI) (Equation 2.1.4.7.1) and entropy (H) (Equation 2.1.4.3.1). Mutual information measures the correlation between feature and class.

$$MI(class, feature) = H(class) - H(Class, feature) \quad \text{(Equation 2.1.4.7.1)}$$

$$MI(class, feature) = H(class, feature) - H(class|feature) - H(feature|class) \quad \text{(Equation 2.1.4.7.2)}$$

The formula of the SU is shown in the Equation 2.1.4.7.3.

$$SU(class, feature) = \frac{2 \times MI(class, feature)}{H(class) + H(feature)} \quad \text{(Equation 2.1.4.7.3)}$$

### 2.1.4.8 Wrapper

The wrapper method uses a selected classifier recursively in cross validation. For each cross validation different combination of features will be used to find a good feature set [49]. The diagram of the wrapper feature selection method can be seen in the Figure 2.1.4.8.1.

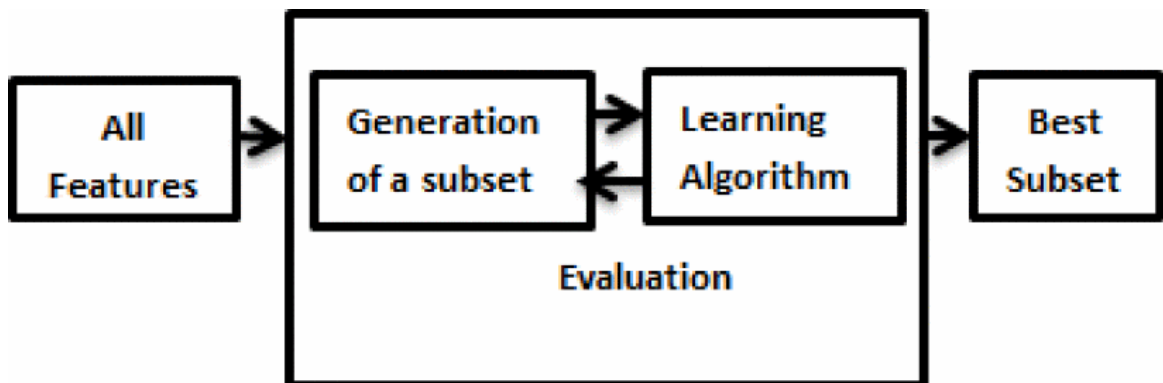


Figure 2.1.4.8.1 Diagram of Wrapper feature selection [49]

## 2.2 Classification Methods

Classification methods contains two main stage. The first stage is finding a generic mathematical model to fit dataset features into classes, the second stage includes the application of the model to new and unseen data in order to predict its class.[50] Following classifier methods are implemented: Random Forest, Logistic Regression, K-Nearest Neighbors, Naïve Bayes, Decision Tree, Support Vector Machines, Adaboost, Radial Basis Function Network, Multilayer Perceptron, Convolution Neural Network.

### 2.2.1 Decision tree learning

In decision trees the features, the decision rules and decisions are represented as nodes, branch and leaf respectively. The idea behind the decision tree learning is to model a tree using various algorithms. The WEKA software uses C4.5 algorithm. The C4.5 algorithm initializes with the original dataset as root. For each iteration, algorithm traverse each feature of the dataset and calculates the information gain (difference in entropy) of the feature. Then it selects the feature which has highest information gain value. Using the selected feature dataset is split. The algorithm is done recursively for each feature thus the decision tree is formed [51][52]. Figure 2.2.1.1 illustrates a simple decision tree. To stop the recursion one of the following cases should be done:

- The elements of the subset belong to same class then the node becomes to a leaf.
- The features are running out, there is no more feature to select, then the most common class is selected and node becomes a leaf.
- The instances are running out, the most common class is selected and node becomes a leaf.

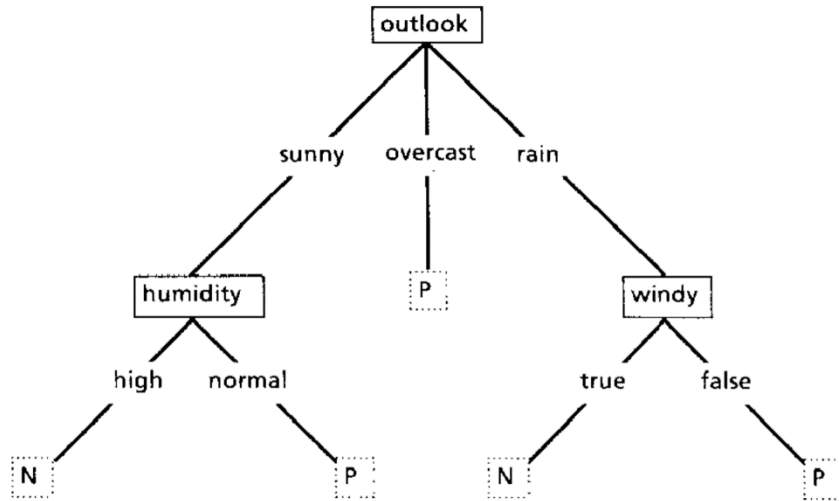


Figure 2.2.1.1 Simple decision tree [52]

### 2.2.2 Random forest

Random forest algorithm creates multiple decision trees using the random sample with replacement algorithm. It uses random and a small part of the training data. Feature bagging is used when trees are constructed. The feature bagging algorithm randomly selects  $\sqrt{n}$  features out of  $n$  features for each split [53]–[55]. The optimum number of trees can be found using the cross-validation. The final decision of the classification is made by majority voting. Figure 2.2.2.1 illustrates a random forest which is constructed from multiple decision trees.

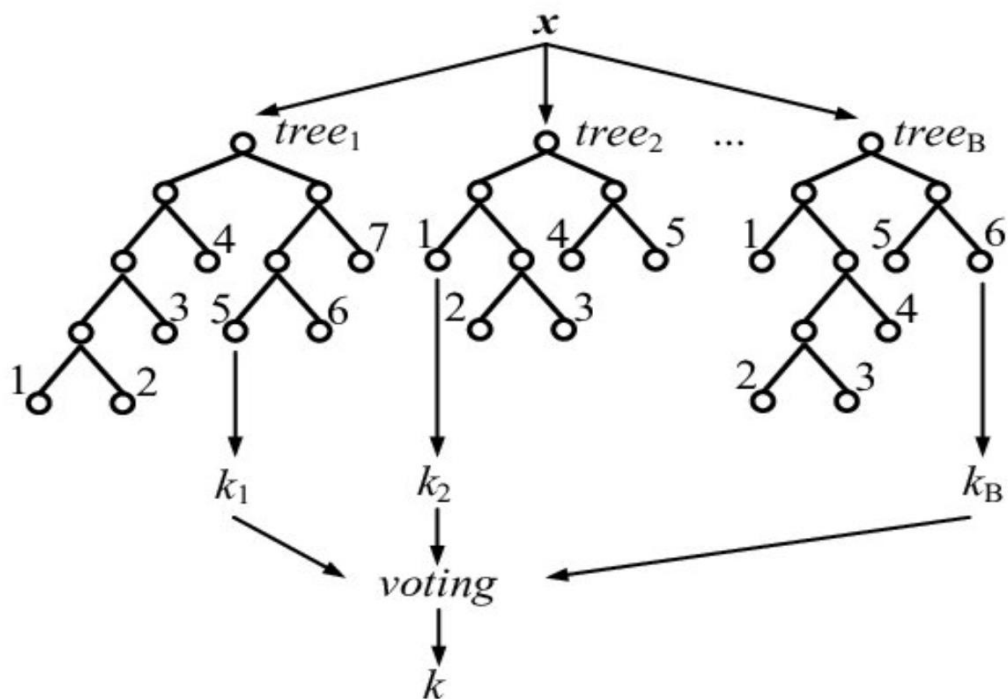


Figure 2.2.2.1 Random forest [56]

### 2.2.3 K-nearest neighbors

The k-NN is an instance-based method. The training part of the algorithm is only storing the entire training set features and its labels. The classification part is made by using the training set directly (no model is needed). To classify a new instance the entire training set is searched and mode of the class value of K similar instances is selected. Figure 2.2.3.1 shows the KNN's working process. The similar instances can be called as neighbors, Euclidian distance (Equation 2.2.3.1) is used for finding the neighbors. [57]–[59]

$$EuclidianDistance(neighbor, instance) = \sqrt{\sum_{n=1}^j (NeighborsFeature_n - InstanceFeature_n)^2} \quad (\text{Equation 2.2.3.1})$$

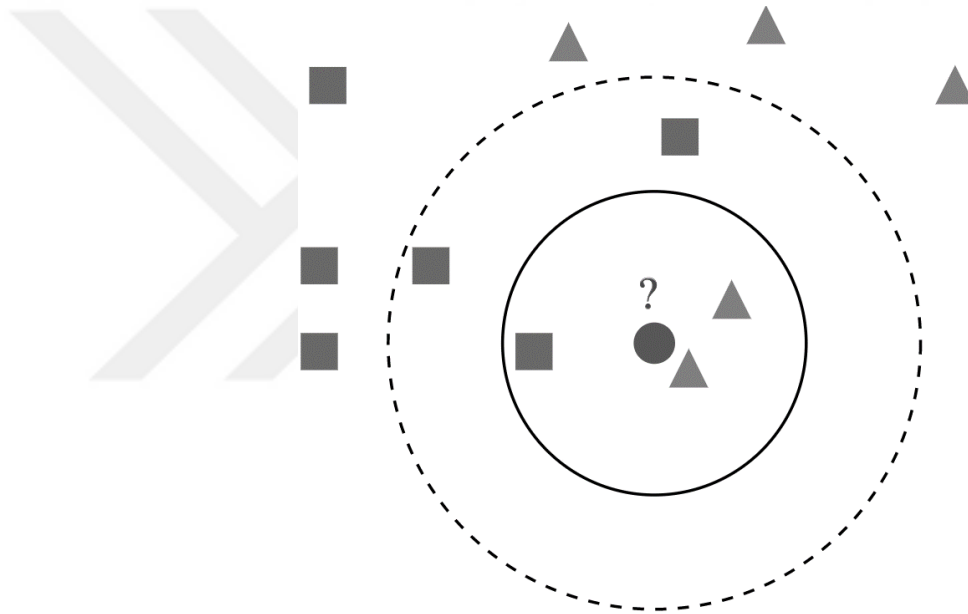


Figure 2.2.3.1 K-nearest neighbors [60]

### 2.2.4 Naïve Bayes classifier

The Naïve Bayes classifier is a probabilistic classifier. It assumes that all the features in are independent. In training part, the probability of each class given different feature values (conditional probability) should be calculated (Equation 2.2.4.1). It is not necessary to fit a model [61][62].

$$P(C_k|x) = \frac{P(C_k) P(x|C_k)}{P(x)} \quad (\text{Equation 2.2.4.1})$$

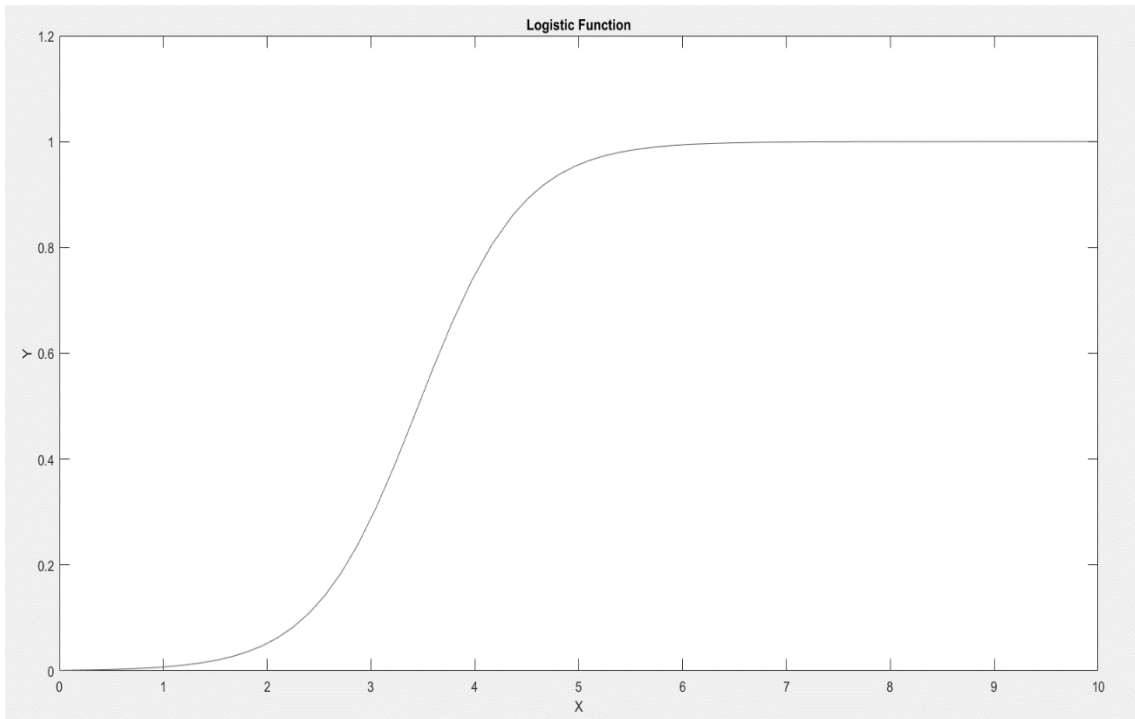


To classify the new unseen data, the predictions can be done using the Bayes theorem (Equation 2.2.4.2) [63].

$$\hat{y} = \underset{k \in \{1, \dots, K\}}{\operatorname{argmax}} p(C_k) \prod_{i=1}^n p(x_i | C_k) \quad (\text{Equation 2.2.4.2})$$

## 2.2.5 Logistic regression

The kernel of the logistic regression is logistic function that also called as sigmoid function. The sigmoid function is shown in Figure 2.2.5.1. It fits an equation that combines coefficients and input values to find output values (Equation 2.2.5.1) [64].



**Figure 2.2.5.1** Logistic Function

$$y = \frac{e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2}}{1 + e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2}} \quad (\text{Equation 2.2.5.1})$$

Where  $\beta_0$  is bias and the  $\beta_1, \beta_2$  values are the coefficients. The coefficients will be estimated using maximum likelihood estimation to minimize the error rate between predicted value and true class using some optimization algorithms [65]. The estimated coefficients are used later for predicting the new cases, if predicted value is approximates to 1, new instance belongs to default class, otherwise it belongs to other class.

## 2.2.6 Support vector machines (SVM)

SVM defines a hyperplane that maximizes the margin between classes using a kernel in training part. Figure 2.2.7.1 shows the visualization of a hyperplane. Numerical optimization techniques are used to find coefficients of the hyperplane. Using hyperplane new data is classified [66], [67].

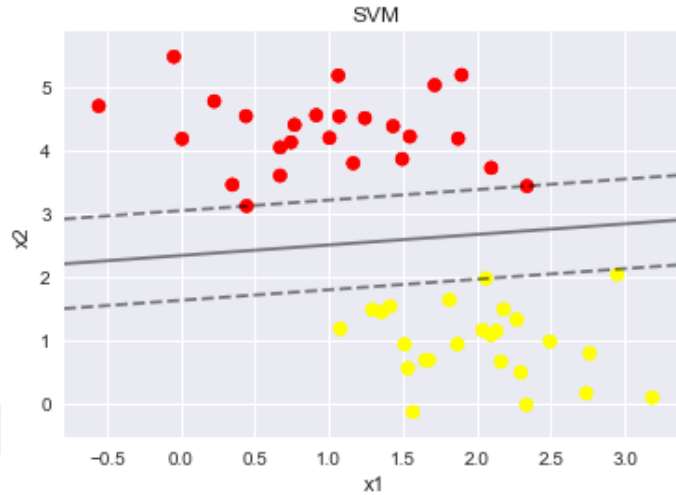


Figure 2.2.7.1 Visualization of hyperplane

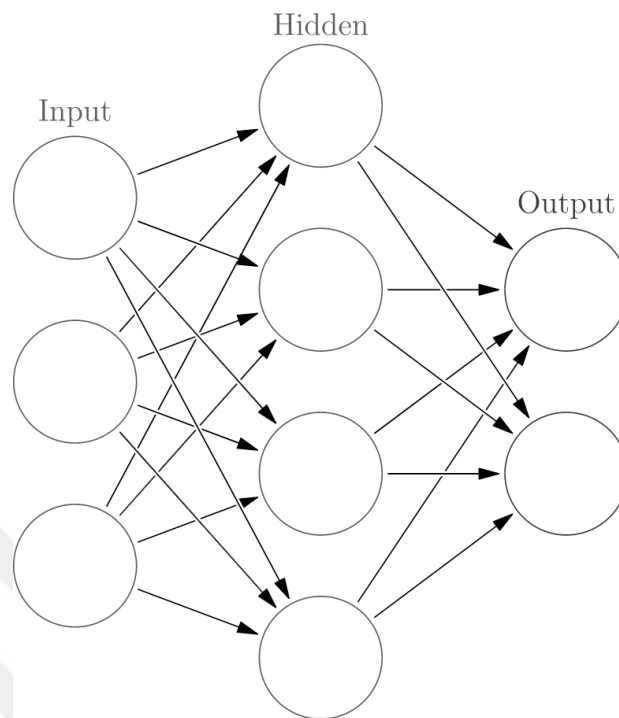
## 2.2.7 Adaptive boosting (AdaBoost)

Adaboost is a meta learner that combines many weak learners in order to build a strong learner [56]. Weak classifiers are trained using a random subset of a training set. Adaboost gives weight to every instance after training part of the weak learner. A misclassified instance gets higher weight so, it appears at the top of the next training subset of next weak classifier. Also, classifiers get a weight based on the accuracy after training as well. To classify a new instance Equation 2.2.7.1 is used where  $h_t(x)$  represents output of weak classifier  $c$  for input  $i$  [68], [69].

$$\hat{y} = \text{sign}\left(\sum_{t=1}^T \text{weight}_{\text{classifier}} \cdot h_t(i)\right) \quad (\text{Equation 2.2.7.1})$$

## 2.2.8 Artificial Neural Network (ANN)

ANN is a framework rather than an algorithm. ANN constructed from connected artificial neurons. Each neuron receives a signal, process it using a non-linear function (activation function) and transmit it to other connected neurons. Neurons and connection between neurons have weights, as the learning process proceeds the weights will be adjusted [70][71]. Figure 2.2.8.1 shows the basic visualization of an ANN.



**Figure 2.2.8.1 Visualization of a ANN [72]**

## **2.2.9 Multilayer Perceptron (MLP)**

MLP is a form of feedforward ANN and it uses backpropagation algorithm for training. MLP consists of an input layer, at least one hidden layer and an output. Each neuron uses nonlinear activation function [70][73]. Figure 2.2.9.1 shows the structure of MLP.

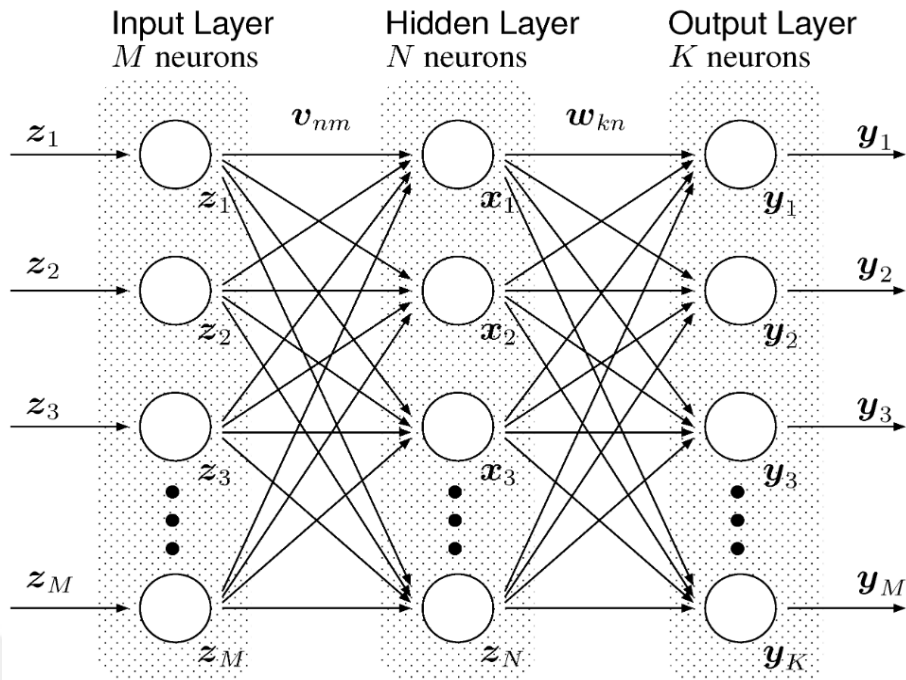


Figure 2.2.9.1 One hidden layer multilayer perceptron [74]

### 2.2.10 Radial basis function network (RBFN)

The RBFN is a type of neural network which uses radial function as activation function (Equation 2.2.10.1) [75], [76]. The network constructed from three layers, input layer, one hidden layer with activation function of radial basis and linear output layer (Figure 2.2.10.1).

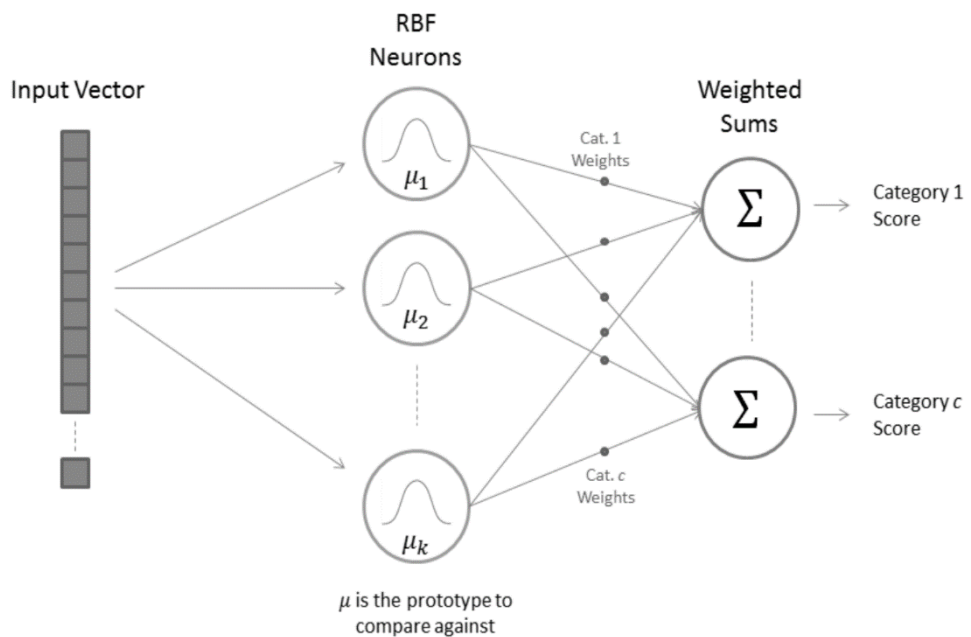


Figure 2.2.10.1 Construction of RBFN [77]

$$\varphi(x) = e^{-\beta\|x-\mu\|^2} \quad \text{(Equation 2.2.10.1)}$$

Where  $\mu$  mean,  $x$  is the input and  $\beta$  is the coefficient which controls the width of the gaussian bell curve of radial function (Figure 2.2.10.2).

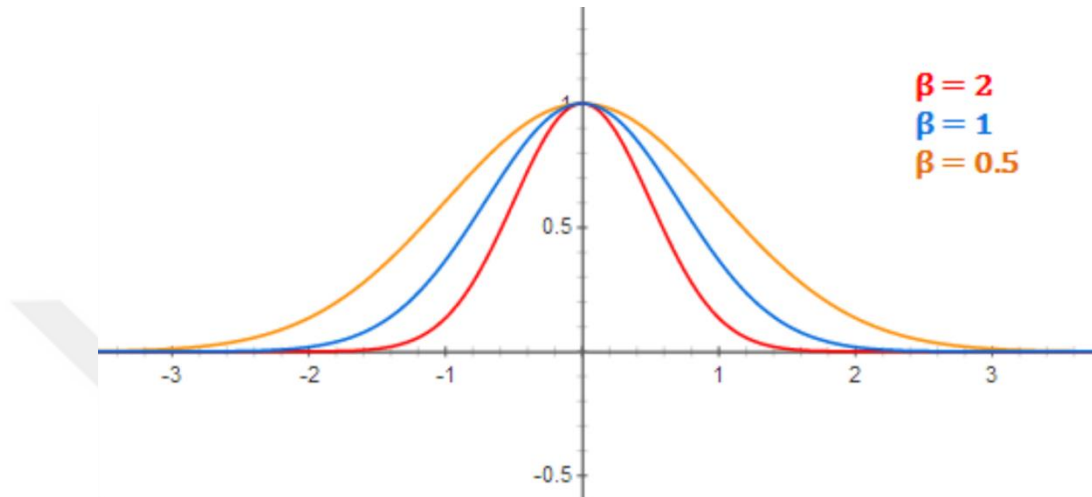


Figure 2.2.10.2 Gaussian radial function with different  $\beta$  values. [77]

## 2.2.11 Convolution neural network (CNN)

CNN is a type of feed-forward artificial neural network which is mostly used for image recognition. The CNN consists of two components which are hidden layer where the feature extraction happens and the classification part which is done by fully connected layers which is shown in Figure 2.2.11.1 [78][79].

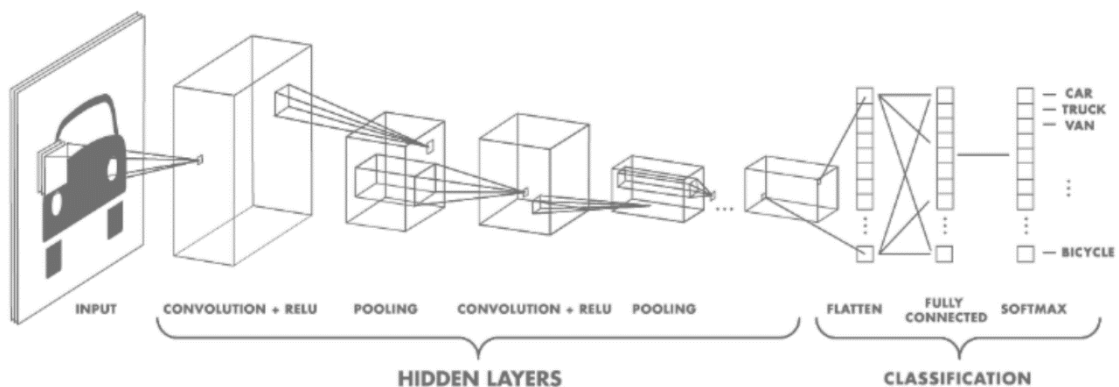


Figure 2.2.11.1 CNN layers [79]

## **2.3 Cross validation techniques (CV)**

CV procedures are used for the estimating the skill of the dataset on a validation set.

### **2.3.1 Ten-fold cross validation (10-fold CV)**

This process is used for the evaluate the machine learning tools skill using resampling procedure [80]. The process is as follows: randomize the dataset, split data set into k fold. For each fold use them as a test set and rest of them are training set and save the score.

### **2.3.2 Leave-one instance out cross validation (LOO-CV)**

In each iteration of the LOOCV, one image is selected as the test data and rest are used as the train set. This process is repeated until all images have been used as the test sample [81].

## **2.4 Classifier evaluation metrics**

Some metrics are used for evaluating the classifiers performance. The metrics that are used in Weka: Accuracy, f-measure, false positive rate, specificity, precision, recall and AUC score. In order to calculate accuracy measures, confusion matrix is constructed.

### **2.4.1 Confusion Matrix**

Confusion matrix visualizes the performance of the algorithm [82]. The rows and the columns of the matrix represents the actual class and predicted class respectively [83]. If there is two class one is labelled as positive (P) and the other is negative (N) the confusion matrix is in the Figure 2.2.2.1.1.

|                     |                              | True condition                       |                                     |
|---------------------|------------------------------|--------------------------------------|-------------------------------------|
|                     |                              | Condition positive                   | Condition negative                  |
| Predicted condition | Total population             |                                      |                                     |
|                     | Predicted condition positive | <b>True positive, Power</b>          | <b>False positive, Type I error</b> |
|                     | Predicted condition negative | <b>False negative, Type II error</b> | <b>True negative</b>                |

Figure 2.2.2.1.1 Confusion Matrix [84]

## 2.4.2 Accuracy

Accuracy (Equation 2.2.2.2.1) can be represented as ratio between number of correctly classified instances and all classified instances [50].

$$Accuracy = \frac{TP+TN}{Number\ of\ instances} \quad \text{(Equation 2.2.2.2.1)}$$

## 2.4.3 Precision

Precision (Equation 2.2.2.3.1) calculates how well the classifier predicted condition positive cases [85].

$$Precision = \frac{TP}{TP+FP} \quad \text{(Equation 2.2.2.3.1)}$$

## 2.4.4 Recall (Sensitivity)

Recall (Equation 2.2.2.4.1) measures the proportion of condition positive cases that classified as condition positive [85].

$$Recall = \frac{TP}{TP+FN} \quad \text{(Equation 2.2.2.4.1)}$$

## 2.4.5 Specificity

Specificity (Equation 2.2.2.5.1) measures the proportion of condition negative cases that classified as condition negative [85].

$$Specificity = \frac{TN}{TN+FP} \quad (\text{Equation 2.2.2.5.1})$$

## 2.4.6 Area under curve score (AUC)

Receiver operating characteristic (ROC) curve shows how a classifier differentiate two different classes. The AUC is the area under ROC curve [85]. Figure 2.2.2.6.1 visualizes a simple ROC curve.

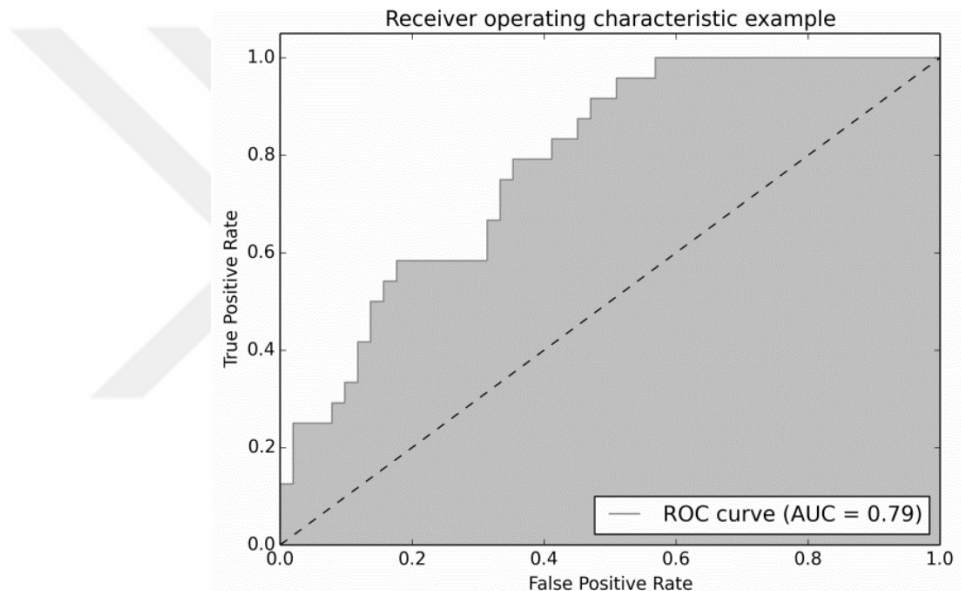


Figure 2.2.2.6.1 Visualization of a ROC curve [86]

## 2.4.7 F1 score

F1 score (Equation 2.2.2.7.1) measures the test accuracy using precision and recall [85].

$$F_1 \text{ score} = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \quad (\text{Equation 2.2.2.7.1})$$

## 2.4.8 False positive rate (FPR)

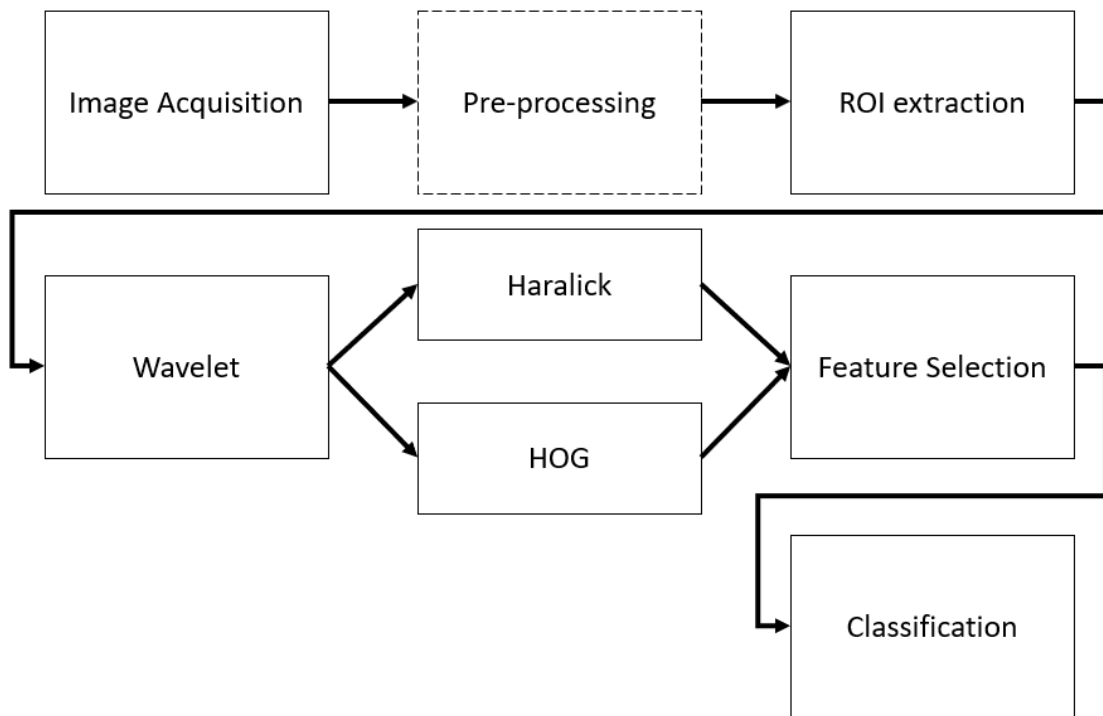
FPR (Equation 2.2.2.8.1) is also called miss rate. It measures the probability of false alarm [85].

$$FPR = \frac{FP}{FP+TN} \quad (\text{Equation 2.2.2.8.1})$$



## 2.5 Proposed Method

Various methods have been tested as explained in Section 2.2. Based on the experimental results given in Section 3.1, a method comprises of the tools that give the best performances is proposed. The summary of the proposed method is shown in Figure 2.3.1 as a flow diagram which has several stages. Explanation of the stages is as follows.

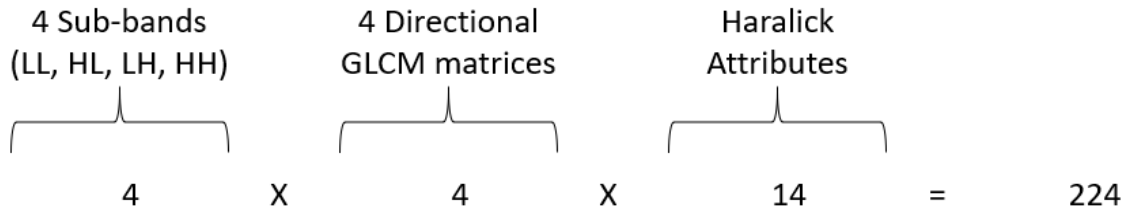


**Figure 2.3.1 Flow Diagram of the Proposed Algorithm**

Diagram starts with image acquisition. The images are acquired from two different datasets (see Section 2.1.1). The experiments of the proposed method are performed for each dataset. As image pre-processing, CLAHE operation is applied on MIAS dataset in order to increase the visibility of the tumorous region (see Figure 3.2.2). However, the CLAHE operation is only applied to MIAS dataset because of the reasons explained in Section 3.2. Pre-processing is followed by a manual ROI cropping (see Section 2.1.1). As a result, the dataset that is used in the algorithm contains cancer positive (marked by an expert) and cancer negative (randomly cropped, see Section 2.1.1) sub-regions. In this thesis those regions are referred as ROIs.

One level Haar wavelet decomposition is applied on the ROI dataset (see Section 2.1.3.1). Haralick features (see Section 2.1.3.3) are extracted from LL, LH, HL, HH sub-

bands of the wavelet decomposed image while, HOG features (see Section 2.1.3.2) are extracted only from LL sub-band. Haralick gives total 224 attributes as explained in Figure 2.3.2.



Before feature selection total number of attributes is  $16 + 224 = 240$ . Most relevant 12 features are selected by Wrapper algorithm (see Section 2.1.4.8). The classifier of the Wrapper algorithm was Random Forest with default parameters.

The selected features are given as an input to Random Forest classifier (see Section 2.2.2). Different number of trees for Random Forest classifier are tested. The optimum number of trees is found as 100.

The proposed method is tested on the datasets. The test results of the proposed method are given in Tables 3.1.50, 3.2.2 and 3.2.7 [87].

# Chapter 3

## Experiments and Analysis

In order to evaluate the performance, various feature extraction and classification methods have been applied on different datasets which is described in the Chapter 2. 10-fold cross validation and leave one instance out cross validation was performed. Hyper parameter optimization has been applied on a validation set. Moreover, using the model which was acquired from cross validations, the breast has been traversed and suspicious areas has been labeled on whole breast. The experiments have been executed using WEKA software and TensorFlow (python) [88], [89].

### 3.1 Evaluation of feature extraction methods

Feature extraction methods converts image into simpler domain which is more understandable for computer, therefore it is important to select a suitable feature extraction method or combination of different feature extraction methods. Various feature extraction methods, which is described in section 2, have been applied and evaluated using different machine learning techniques on Pilot dataset. Table 3.1.1 shows 10-fold cross validation using Random forest classifier applied on dataset without normalization. The random forests number of trees is set to 100 which is the default value of WEKA. Class 0 represents normal classified instances, class 1 represents abnormal instances. Accuracy metrics are represented in the tables. Moreover, to achieve a great performance accuracy metrics of both classes should be higher.

| Metrics(%<br>)<br>\Class | 0    | 1    | Weighted<br>Avg. |
|--------------------------|------|------|------------------|
| TP Rate                  | 71.0 | 77.4 | 74.2             |
| FP Rate                  | 22.6 | 29   | 25.8             |
| Precision                | 75.9 | 72.7 | 74.3             |
| Recall                   | 71.0 | 77.4 | 74.2             |
| F-Measure                | 73.3 | 75   | 74.2             |
| ROC Area                 | 80.6 | 80.6 | 8.06             |
| Overall<br>Accuracy      |      |      | 77               |

**Table 3.1.1 Haralick features are extracted from Pilot dataset and evaluated using 10-fold cross validation with random forest classifier without normalization**

The experiment which is described above has been repeated using different sized ROI datasets. The datasets are generated using the such methodology that fixes number of the abnormal instances and increases the number of the normal cases by two times, three times, four times, five times of the number of normal instances. Thus, there is four more datasets. For instance, biggest ROI dataset of Pilot dataset has 186 instances (31 abnormal, 155 normal). Table 3.1.2 shows the 10-fold cross validation accuracy metrics using random forest classifier on dataset which contains 31 abnormal 62 normal instances. The table 3.1.2 shows better evaluation metrics than the Table 3.1.1 for class 0 and weighted average of all cases. On the other hand, class one's (malignant cases) TP Rate and f-measure has been dropped.

| Metrics(%)\Class | 0    | 1    | Weighted Avg. |
|------------------|------|------|---------------|
| TP Rate          | 85.5 | 67.7 | 79.6          |
| FP Rate          | 32.3 | 14.5 | 26.3          |
| Precision        | 84.1 | 70.0 | 79.4          |
| Recall           | 85.5 | 67.7 | 79.6          |
| F-Measure        | 84.8 | 68.9 | 79.5          |
| ROC Area         | 83.2 | 83.2 | 83.2          |
| Overall Accuracy |      |      | 79            |

**Table 3.1.2 Haralick features are extracted from the dataset which is constructed from 31 abnormal and 62 normal instances and evaluated using 10-fold CV with random forest**

Table 3.1.3 shows 10-fold CV results of another ROI dataset that has 31 abnormal 93 abnormal instances. Compared to Table 3.1.1 the performance metrics drops for class 1 and increases for class 0 also weighted average of the metrics are decreased.

| Metrics(%)\Class | 0    | 1    | Weighted Avg. |
|------------------|------|------|---------------|
| TP Rate          | 95.7 | 12.9 | 75            |
| FP Rate          | 87.1 | 4.3  | 66.4          |
| Precision        | 76.7 | 50   | 70            |
| Recall           | 95.7 | 12.9 | 75            |
| F-Measure        | 85.2 | 20.5 | 69            |
| ROC Area         | 54.4 | 54.4 | 54.4          |
| Overall Accuracy |      |      | 75            |

**Table 3.1.3 Haralick features are extracted from the dataset which is constructed from 31 abnormal and 93 normal instances and evaluated using 10-fold CV with random forest**

Table 3.1.4 and 3.1.5 corresponds to evaluation metrics of 10-fold CV for data sets that have 31 abnormal 124 normal and 31 abnormal 155 normal instances respectively.

| Metrics(%)\Class | 0    | 1    | Weighted Avg. |
|------------------|------|------|---------------|
| TP Rate          | 94.4 | 12.9 | 78.1          |
| FP Rate          | 87.1 | 5.6  | 70.8          |
| Precision        | 81.3 | 36.4 | 72.3          |
| Recall           | 94.4 | 12.9 | 78.1          |
| F-Measure        | 87.3 | 19   | 73.7          |
| ROC Area         | 51.8 | 51.8 | 51.8          |
| Overall Accuracy |      |      | 78            |

**Table 3.1.4 Haralick features are extracted from the dataset which contains 31 abnormal and 124 normal instances and evaluated using 10-fold CV with random forest**

| Metrics(%)\Class | 0    | 1    | Weighted Avg. |
|------------------|------|------|---------------|
| TP Rate          | 95.5 | 3.2  | 80.1          |
| FP Rate          | 96.8 | 4.5  | 81.4          |
| Precision        | 83.1 | 12.5 | 71.4          |
| Recall           | 95.5 | 3.2  | 80.1          |
| F-Measure        | 88.9 | 5.1  | 74.9          |
| ROC Area         | 65.1 | 65.1 | 65.1          |
| Overall Accuracy |      |      | 78            |

**Table 3.1.5 Haralick features are extracted from the dataset which contains 31 abnormal and 155 normal instances and evaluated using 10-fold CV with random forest**

According to table 3.1.4 and 3.1.5 accuracy metrics of abnormal cases are increased because of the dataset size rise. But it is kind of overfitting because while normal cases accuracy metrics increasing, the abnormal cases metrics decreasing critically, the classifier predicts every instance as normal which is predominant.

Haralick features have been normalized on balanced dataset and imbalanced dataset which contains 31 abnormal 62 normal cases. The results are shown in table 3.1.6 and 3.1.7. In table 3.1.1 and 3.1.2, 3.1.3 and 3.1.5 the accuracy metrics are almost same except ROC area. The features which are not normalized perform better than the normalized version. The upcoming experiments are done using balanced dataset and one of the imbalanced datasets which contains 31 normal 62 abnormal instances.

| Metrics(%<br>)\Class | 0    | 1    | Weighted<br>Avg. |
|----------------------|------|------|------------------|
| TP Rate              | 71   | 77.4 | 74.2             |
| FP Rate              | 22.6 | 29   | 25.8             |
| Precision            | 75.9 | 72.7 | 74.3             |
| Recall               | 71   | 77.4 | 74.2             |
| F-Measure            | 73.3 | 75   | 74.2             |
| ROC Area             | 79.7 | 79.7 | 79.7             |
| Overall<br>Accuracy  |      |      | 80               |

**Table 3.1.6 Haralick features are extracted from balanced dataset, normalized and evaluated using 10-fold cross validation with random forest classifier**

| Metrics(%)\Class | 0    | 1    | Weighted Avg. |
|------------------|------|------|---------------|
| TP Rate          | 88.7 | 64.5 | 80.6          |
| FP Rate          | 35.5 | 11.3 | 27.4          |
| Precision        | 83.3 | 74.1 | 80.2          |
| Recall           | 88.7 | 64.5 | 80.6          |
| F-Measure        | 85.9 | 69   | 80.3          |
| ROC Area         | 86   | 86   | 86            |
| Overall Accuracy |      |      | 80            |

**Table 3.1.7 Haralick features are extracted from unbalanced dataset, normalized and evaluated using 10-fold cross validation with random forest classifier**

Other classification methods are used without normalization. Table 3.1.8 and 3.1.9 shows Haralick feature extraction with AdaBoost classifier on balanced and imbalanced dataset respectively. The number of iterations of the AdaBoost classifier was 10. According to table 3.1.8 and 3.1.9 Random forest gives slightly better results than Adaboost. Table 3.1.9 shows that balanced dataset gives better results than the imbalanced dataset in case of Adaboost.



| Metrics(%)\Class | 0    | 1    | Weighted Avg. |
|------------------|------|------|---------------|
| TP Rate          | 71   | 77.4 | 74.2          |
| FP Rate          | 22.6 | 29   | 25.8          |
| Precision        | 75.9 | 72.7 | 74.3          |
| Recall           | 71   | 77.4 | 74.2          |
| F-Measure        | 73.3 | 75   | 74.2          |
| ROC Area         | 78.6 | 78.6 | 78.6          |
| Overall Accuracy |      |      | 74            |

**Table 3.1.8 Haralick features are extracted from balanced dataset evaluated using 10-fold cross validation using AdaBoost classifier**

| Metrics(%)\Class | 0    | 1    | Weighted Avg. |
|------------------|------|------|---------------|
| TP Rate          | 82.3 | 64.5 | 76.3          |
| FP Rate          | 35.5 | 17.7 | 29.6          |
| Precision        | 82.3 | 64.5 | 76.3          |
| Recall           | 82.3 | 64.5 | 76.3          |
| F-Measure        | 82.3 | 64.5 | 76.3          |
| ROC Area         | 82   | 82   | 82            |
| Overall Accuracy |      |      | 76            |

**Table 3.1.9. Haralick features are extracted from imbalanced dataset evaluated using 10-fold cross validation using AdaBoost classifier**

Table 3.1.10 and 3.1.11 present the accuracy metrics of KNN classifier. Parameter of KNN were set to default values which is 1 for K value. KNN classifier gives close results to Adaboost and Random forest for both balanced and imbalanced dataset but still Random Forest gives best results

| Metrics(%<br>)\Class | 0    | 1    | Weighted<br>Avg. |
|----------------------|------|------|------------------|
| TP Rate              | 67.7 | 77.4 | 72.6             |
| FP Rate              | 22.6 | 32.3 | 27.4             |
| Precision            | 75   | 70.6 | 72.8             |
| Recall               | 67.7 | 77.4 | 72.6             |
| F-Measure            | 71.2 | 73.8 | 72.5             |
| ROC Area             | 74.5 | 74.5 | 74.5             |
| Overall<br>Accuracy  |      |      | 72               |

**Table 3.1.10** Haralick features are extracted from balanced dataset evaluated using 10-fold cross validation using KNN classifier

| Metrics(%<br>)\Class | 0    | 1    | Weighted<br>Avg |
|----------------------|------|------|-----------------|
| TP Rate              | 80.6 | 64.5 | 75.3            |
| FP Rate              | 35.5 | 19.4 | 30.1            |
| Precision            | 82   | 62.5 | 75.5            |
| Recall               | 80.6 | 64.5 | 75.3            |
| F-Measure            | 81.3 | 63.5 | 75.4            |
| ROC Area             | 72.7 | 72.7 | 72.7            |
| Overall<br>Accuracy  |      |      | 75              |

**Table 3.1.11** Haralick features are extracted from imbalanced dataset evaluated using 10-fold cross validation using KNN classifier

Logistic Regression is used as classifier the results are shown in Table 3.1.12 and 3.1.13. Parameters of Logistic Regression were set to default values. The accuracy metrics are better than other classifiers when the imbalanced data is used while, the balanced data results are approximately same.

| Metrics(%)\Class | 0    | 1    | Weighted Avg. |
|------------------|------|------|---------------|
| TP Rate          | 64.5 | 77.4 | 71            |
| FP Rate          | 22.6 | 35.5 | 29            |
| Precision        | 74.1 | 68.6 | 71.3          |
| Recall           | 64.5 | 77.4 | 71            |
| F-Measure        | 69   | 72.7 | 70.8          |
| ROC Area         | 67.6 | 73.7 | 70.7          |
| Overall Accuracy |      |      | 70            |

**Table 3.1.12 Haralick features are extracted from balanced dataset evaluated using 10-fold cross validation using Logistic Regression classifier**

| Metrics(%)\Class | 0    | 1    | Weighted Avg. |
|------------------|------|------|---------------|
| TP Rate          | 85.5 | 77.4 | 82.8          |
| FP Rate          | 22.6 | 14.5 | 19.9          |
| Precision        | 88.3 | 72.7 | 83.1          |
| Recall           | 85.5 | 77.4 | 82.8          |
| F-Measure        | 86.9 | 75   | 82.9          |
| ROC Area         | 86.8 | 87.1 | 86.9          |
| Overall Accuracy |      |      | 82            |

**Table 3.1.13. Haralick features are extracted from unbalanced dataset evaluated using 10-fold cross validation using Logistic Regression classifier**

Naïve Bayes classifier is used, the results are described in table 3.1.14 and 3.1.15. Parameters were set to default values. The classifier gives approximately similar results for abnormal cases, on balanced dataset.

| Metrics(%)\Class | 0    | 1    | Weighted Avg. |
|------------------|------|------|---------------|
| TP Rate          | 51.6 | 87.1 | 69.4          |
| FP Rate          | 12.9 | 48.4 | 30.6          |
| Precision        | 80   | 64.3 | 72.1          |
| Recall           | 51.6 | 87.1 | 69.4          |
| F-Measure        | 62.7 | 74   | 68.4          |
| ROC Area         | 77.9 | 78.5 | 78.2          |
| Overall Accuracy |      |      | 69            |

**Table 3.1.14 Haralick features are extracted from balanced dataset evaluated using 10-fold cross validation using Naive Bayes classifier**

| Metrics(%)\Class | 0    | 1    | Weighted Avg. |
|------------------|------|------|---------------|
| TP Rate          | 45.2 | 93.5 | 61.3          |
| FP Rate          | 6.5  | 54.8 | 22.6          |
| Precision        | 93.3 | 46   | 77.6          |
| Recall           | 45.2 | 93.5 | 61.3          |
| F-Measure        | 60.9 | 61.7 | 61.1          |
| ROC Area         | 85   | 79.6 | 83.2          |
| Overall Accuracy |      |      | 61            |

**Table 3.1.15 Haralick features are extracted from unbalanced dataset evaluated using 10-fold cross validation using Naive Bayes classifier**

Table 3.1.16 and 3.1.17 shows the results of Decision Tree classification method. The results are similar to Random Forest classifiers results.

| Metrics(%)\Class | 0    | 1    | Weighted Avg. |
|------------------|------|------|---------------|
| TP Rate          | 67.7 | 64.5 | 66.1          |
| FP Rate          | 35.5 | 32.3 | 33.9          |
| Precision        | 65.6 | 66.7 | 66.1          |
| Recall           | 67.7 | 64.5 | 66.1          |
| F-Measure        | 66.7 | 65.6 | 66.1          |
| ROC Area         | 66.5 | 66.5 | 66.5          |
| Overall Accuracy |      |      | 67            |

**Table 3.1.16** Haralick features are extracted from balanced dataset evaluated using 10-fold cross validation using Decision Tree classifier

| Metrics(%)\Class | 0    | 1     | Weighted Avg. |
|------------------|------|-------|---------------|
| TP Rate          | 80.6 | 61.3  | 74.2          |
| FP Rate          | 38.7 | 19.4  | 32.3          |
| Precision        | 80.6 | 61.3  | 74.2          |
| Recall           | 80.6 | 61.3  | 74.2          |
| F-Measure        | 80.6 | 61.3. | 74.2          |
| ROC Area         | 70.8 | 70.8  | 70.8          |
| Overall Accuracy |      |       | 74            |

**Table 3.1.17** Haralick features are extracted from imbalanced dataset evaluated using 10-fold cross validation using Decision Tree classifier

SVM is another classification algorithm that is used to evaluate the performance of the Haralick features. Parameters of the SVM classifier was default values. Table 3.1.18 and 3.1.19 demonstrates the results of unbalanced and balanced dataset. Performance of the SVM is lower than previous methods. Especially when the unbalanced dataset is used the classifier overfits and classifies every instance as predominant class's label.

| Metrics(%<br>)\Class | 0    | 1    | Weighted<br>Avg. |
|----------------------|------|------|------------------|
| TP Rate              | 87.1 | 9.7  | 48.4             |
| FP Rate              | 90.3 | 12.9 | 51.6             |
| Precision            | 49.1 | 42.9 | 46               |
| Recall               | 87.1 | 9.7  | 48.4             |
| F-<br>Measure        | 62.8 | 15.8 | 39.3             |
| ROC<br>Area          | 48.  | 48.4 | 48.4             |
| Overall<br>Accuracy  |      |      | 48.4             |

**Table 3.1.18** Haralick features are extracted from balanced dataset evaluated using 10-fold cross validation using SVM classifier

| Metrics(%<br>)\Class | 0   | 1   | Weighted<br>Avg. |
|----------------------|-----|-----|------------------|
| TP Rate              | 100 | 0   | 66.7             |
| FP Rate              | 100 | 100 | 66.7             |
| Precision            | 667 | 0   | 44.4             |
| Recall               | 100 | 0   | 66.7             |
| F-<br>Measure        | 80  | 0   | 53.3             |
| ROC<br>Area          | 50  | 50  | 50               |
| Overall<br>Accuracy  |     |     | 66               |

**Table 3.1.19** Haralick features are extracted from unbalanced dataset evaluated using 10-fold cross validation using SVM classifier

Table 3.1.20 and 3.1.21 shows the results of the RBF-NN on both datasets. The kernel of the RBF-NN was random forest, and the parameters was set to default values. The classifier gives almost similar results but table X24 demonstrates that the second classes f1 score is decreased when it compared to balanced dataset.

| Metrics(% )\Class | 0    | 1    | Weighted Avg. |
|-------------------|------|------|---------------|
| TP Rate           | 48.4 | 71   | 59.7          |
| FP Rate           | 29   | 51.6 | 40.3          |
| Precision         | 62.5 | 57.9 | 60.2          |
| Recall            | 48.4 | 71   | 59.7          |
| F-Measure         | 54.5 | 63.8 | 59.2          |
| ROC Area          | 59.4 | 59.2 | 59.3          |
| Overall Accuracy  |      |      | 59            |

**Table 3.1.20. Haralick features are extracted from balanced dataset evaluated using 10-fold cross validation using RBF-NN classifier**

| Metrics(% )\Class | 0    | 1    | Weighted Avg. |
|-------------------|------|------|---------------|
| TP Rate           | 74.2 | 61.3 | 69.9          |
| FP Rate           | 38.7 | 25.8 | 34.4          |
| Precision         | 79.3 | 54.3 | 71            |
| Recall            | 74.2 | 61.3 | 69.9          |
| F-Measure         | 76.7 | 57.6 | 70.3          |
| ROC Area          | 74.7 | 74.7 | 74.7          |
| Overall Accuracy  |      |      | 69.9          |

**Table 3.1.21 Haralick features are extracted from unbalanced dataset evaluated using 10-fold cross validation using RBF-NN classifier**

LBP features has been extracted and evaluated using the random forest and 10-fold CV, the table 3.1.22 shows the accuracy metrics. This experiment is repeated using normalized dataset, which is shown in table 3.1.23. The normalization does not change accuracy metrics. LBP features and Random forest combination does not perform well in comparison with Haralick and random forest.

| Metrics(\%)\Class | 0    | 1    | Weighted Avg. |
|-------------------|------|------|---------------|
| TP Rate           | 22.6 | 54.8 | 38.7          |
| FP Rate           | 45.2 | 77.4 | 61.3          |
| Precision         | 33.3 | 41.5 | 37.4          |
| Recall            | 22.6 | 54.8 | 38.7          |
| F-Measure         | 26.9 | 47.2 | 37.1          |
| ROC Area          | 31.9 | 31.9 | 31.9          |
| Overall Accuracy  |      |      | 38            |

**Table 3.1.22 LBP features are extracted from balanced dataset, evaluated using 10-fold cross validation with random forest classifier**

| Metrics(\%)\Class | 0    | 1    | Weighted Avg. |
|-------------------|------|------|---------------|
| TP Rate           | 22.6 | 54   | 38.7          |
| FP Rate           | 45.2 | 77.4 | 61.3          |
| Precision         | 33.3 | 41.5 | 37.4          |
| Recall            | 22.6 | 54.8 | 38.7          |
| F1-Measure        | 26.9 | 47.2 | 37.1          |
| ROC Area          | 31.9 | 31.9 | 31.9          |
| Overall Accuracy  |      |      | 38            |

**Table 3.1.23 LBP features are extracted from balanced dataset, normalized and evaluated using 10-fold cross validation with random forest classifier**

LBP features are extracted from balanced and unbalanced dataset the results are shown in table 3.1.24 and 3.1.25. The results of unbalanced and balanced dataset show that the classifier works better with Haralick features. Moreover, both Haralick and LBP features on balanced and imbalanced datasets shows that normalization decreases the class 1's accuracy measures and overall accuracy.



| Metrics(%)\Class | 0    | 1    | Weighted Avg. |
|------------------|------|------|---------------|
| TP Rate          | 85.5 | 19.4 | 63.4          |
| FP Rate          | 80.6 | 14.5 | 58.6          |
| Precision        | 67.9 | 40   | 58.6          |
| Recall           | 85.5 | 19.4 | 63.4          |
| F-Measure        | 75.7 | 26.1 | 59.2          |
| ROC Area         | 57.1 | 57.1 | 57.1          |
| Overall Accuracy |      |      | 63            |

**Table 3.1.24 LBP features are extracted from balanced dataset, evaluated using 10-fold cross validation random forest classifier**

| Metrics(%)\Class | 0    | 1    | Weighted Avg. |
|------------------|------|------|---------------|
| TP Rate          | 83.9 | 16.1 | 61.3          |
| FP Rate          | 83.9 | 16.1 | 61.3          |
| Precision        | 66.7 | 33.3 | 55.6          |
| Recall           | 83.9 | 16.1 | 61.3          |
| F-Measure        | 74.3 | 21.7 | 56.8          |
| ROC Area         | 53.3 | 53.3 | 53.3          |
| Overall Accuracy |      |      | 61            |

**Table 3.1.25 LBP features are extracted from unbalanced dataset, normalized and evaluated using 10-fold cross validation random forest classifier**

Since the normalization does not changes any accuracy measures, even worsens them, from now on the experiments done on unnormalized dataset. LBP features are tested using AdaBoost classifier the results are showed on table 3.1.26 and 3.1.27.

AdaBoost does not work well on LBP features. Adaboost performance was better on the Haralick features.

| Metrics(\%)\Class | 0    | 1    | Weighted Avg. |
|-------------------|------|------|---------------|
| TP Rate           | 41.9 | 58.1 | 50            |
| FP Rate           | 41.9 | 58.1 | 50            |
| Precision         | 50   | 50   | 50            |
| Recall            | 41.9 | 58.1 | 50            |
| F-Measure         | 45.6 | 53.7 | 49.7          |
| ROC Area          | 41.5 | 41.5 | 41.5          |
| Overall Accuracy  |      |      | 50            |

**Table 3.1.26 LBP features are extracted from balanced dataset and evaluated using 10-fold cross validation AdaBoost classifier**

| Metrics(\%)\Class | 0    | 1    | Weighted Avg. |
|-------------------|------|------|---------------|
| TP Rate           | 82.3 | 19.4 | 61.3          |
| FP Rate           | 80.6 | 17.7 | 59.7          |
| Precision         | 67.1 | 35.3 | 56.5          |
| Recall            | 82.3 | 19.4 | 61.3          |
| F-Measure         | 73.9 | 25   | 57.6          |
| ROC Area          | 52.1 | 52.1 | 52.1          |
| Overall Accuracy  |      |      | 61            |

**Table 3.1.27. LBP features are extracted from imbalanced dataset and evaluated using 10-fold cross validation AdaBoost classifier**

The combination of LBP features and AdaBoost classifier has been tried on imbalanced dataset, table 3.1.27 shows the results. The overall accuracy seems higher than the balanced dataset results however, the class 1's accuracy metrics decreased.

For imbalanced and balanced datasets, the f-measure gives the overall test accuracy. The appropriate measure to test our algorithms accuracy is to use class 1's f-

measure. Overall f-measure is not suitable to test the performance of the algorithm as shown previously for imbalanced datasets, it biases to the class 0's f-measure. Table 3.1.28 shows classifiers performance on LBP features and class 1's f-measures for balanced and imbalanced dataset.

| Classifier/F-measures (%) | Balanced Dataset | Imbalanced Dataset |
|---------------------------|------------------|--------------------|
| K-NN                      | 40.0             | 23.7               |
| Logistic regression       | <b>63.2</b>      | 62.9               |
| RBF Network               | 46.4             | 25.0               |
| Naïve Bayes               | 47.2             | 37.1               |
| Decision Tree             | 51.4             | 25.5               |
| SVM                       | 43.3             | 0                  |

**Table 3.1.28 LBP features are extracted from balanced and imbalanced dataset, evaluated using 10-fold cross validation with different classifiers**

Table 3.1.28 shows the LBP features test results which was done by various classifiers on different sized datasets. The performance of the LBP feature extraction method is not enough when it is compared with the Haralick's features. LBP feature works well on balanced dataset with Logistic Regression classifier with default parameters. The overall accuracy for this case is 66%. The detailed measures are shown in table 3.1.29.

| Metrics(%)\Class | 0    | 1    | Weighted Avg. |
|------------------|------|------|---------------|
| TP Rate          | 58.1 | 74.2 | 66.1          |
| FP Rate          | 25.8 | 41.9 | 33.9          |
| Precision        | 69.2 | 63.9 | 66.6          |
| Recall           | 58.1 | 74.2 | 66.1          |
| F-Measure        | 63.2 | 68.7 | 65.9          |
| ROC Area         | 74.3 | 73.5 | 73.9          |
| Overall Accuracy |      |      | 66            |

**Table 3.1.29 LBP features are extracted from balanced dataset, evaluated using 10-fold cross validation with Logistic Regression classifier**

TAS features are extracted from both imbalanced and balanced dataset. Same process that is applied to LBP and Haralick features is repeated using TAS feature. Table 3.1.30 shows the experiment results of the class 1's. The best f-measure of TAS features evaluated using Naïve Bayes classifier this is followed by Logistic Regression and SVM classifiers. The detailed accuracy measures of Naïve Bayes Classifier are shown in table 3.1.31.

| Classifier/F-measures (%) | Balanced Dataset | Imbalanced Dataset |
|---------------------------|------------------|--------------------|
| Random Forest             | 55.7             | 60.4               |
| Adaboost                  | 62.7             | 57.1               |
| K-NN                      | 59.7             | 59.4               |
| Logistic regression       | <b>64.6</b>      | 52.8               |
| RBF Network               | 58.1             | 54.8               |
| Naïve Bayes               | <b>66.7</b>      | 59.5               |
| Decision Tree             | 46.4             | 56.7               |
| SVM                       | <b>64.4</b>      | 57.1               |

**Table 3.1.30 TAS features are extracted from balanced and imbalanced dataset, evaluated using 10-fold cross validation with different classifiers**

| Metrics(%)\Class | 0    | 1    | Weighted Avg. |
|------------------|------|------|---------------|
| TP Rate          | 51.6 | 74.2 | 62.9          |
| FP Rate          | 25.8 | 48.4 | 37.1          |
| Precision        | 66.7 | 60.5 | 63.6          |
| Recall           | 51.6 | 74.2 | 62.9          |
| F-Measure        | 58.2 | 66.7 | 62.4          |
| ROC Area         | 67   | 67.2 | 67.1          |
| Overall Accuracy |      |      | 62            |

**Table 3.1.31 TAS features are extracted from balanced dataset, evaluated using 10-fold cross validation with Naive Bayes classifier**

Zernike features are extracted and evaluated using the methodology that have been done previously. Table 3.1.32 shows the f-measure of class 1 using different classifiers.

| Classifier/F-measures (%) | Balanced Dataset | Imbalanced Dataset |
|---------------------------|------------------|--------------------|
| Random Forest             | <b>55.2</b>      | 18.2               |
| Adaboost                  | 44.8             | 39.3               |
| K-NN                      | <b>56.3</b>      | 44.8               |
| Logistic regression       | 42.4             | 33.3               |
| RBF Network               | 46.2             | 50.0               |
| Naïve Bayes               | <b>55.1</b>      | 50.0               |
| Decision Tree             | 50.0             | 30.2               |
| SVM                       | 50.7             | 0                  |

**Table 3.1.32 Zernike features are extracted from balanced and imbalanced dataset, evaluated using 10-fold cross validation with different classifiers**

Zernike features works well with K-NN classifier on balanced dataset. The performance of the Zernike features close to LBP features and worse than the Haralick and TAS features. The SVM classifier is gives the worst result for the class 1 on unbalanced dataset because of overfitting.

Detailed performance metrics of the K-NN and Zernike features are described in the table 3.1.33. The parameters of the K-NN is set to default values.

| Metrics(%)\Class | 0    | 1    | Weighted Avg. |
|------------------|------|------|---------------|
| TP Rate          | 35.5 | 64.5 | 50            |
| FP Rate          | 35.5 | 64.5 | 50            |
| Precision        | 50   | 50   | 50            |
| Recall           | 35.5 | 64.5 | 50            |
| F-Measure        | 41.5 | 56.3 | 48.9          |
| ROC Area         | 46.8 | 46.8 | 46.8          |
| Overall Accuracy |      |      | 50            |

**Table 3.1.33 Zernike features are extracted from balanced dataset, evaluated using 10-fold cross validation with K-NN classifier**

From GLCM matrix 6 statistical features which are energy, homogeneity, correlation, dissimilarity, ASM and contrast have been extracted. The f-measures of class 1 is shown in table 3.1.34. Best performance is achieved when RBF Network is used as classifier. SVM classifier, same as the Zernike features, overfits to data on imbalanced dataset. Classify all instances as class 0 which has more instances.

| Classifier/F-measures (%) | Balanced Dataset | Imbalanced Dataset |
|---------------------------|------------------|--------------------|
| Random Forest             | 69.7             | 58.6               |
| Adaboost                  | 66.7             | 64.3               |
| K-NN                      | 63.5             | 61.5               |
| Logistic regression       | 76.9             | 73.8               |
| RBF Network               | <b>78.8</b>      | 36.4               |
| Naïve Bayes               | 70.0             | 54.9               |
| Decision Tree             | 64.7             | 58.0               |
| SVM                       | 74.0             | 0                  |

**Table 3.1.34 GLCM matrix statistical features are extracted from balanced and imbalanced dataset, evaluated using 10-fold cross validation with different classifiers**

Other metrics than the f-measure of RBF Network is shown in table 3.1.35. This feature extraction technique works better than the previous TAS, Zernike and LBP. Combination of GLCM statistical features and RBF Network gives slightly higher results than the Haralick's and Random Forest combination.

| Metrics(%)\Class | 0    | 1           | Weighted Avg. |
|------------------|------|-------------|---------------|
| TP Rate          | 71   | 83.9        | 77.4          |
| FP Rate          | 16.1 | 29          | 22.6          |
| Precision        | 81.5 | 74.3        | 77.9          |
| Recall           | 71   | 83.9        | 77.4          |
| F-Measure        | 75.9 | <b>78.8</b> | 77.3          |
| ROC Area         | 73   | 73.3        | 73.2          |
| Overall Accuracy |      |             | <b>77</b>     |

**Table 3.1.35 GLCM matrix statistical features are extracted from balanced dataset, evaluated using 10-fold cross validation with RBF Network classifier**

Hog features are extracted and evaluated using different classifiers. Table 3.1.36 give the f-measure of the abnormal cases. Naive Bayes and HOG features combination gives better result than the other classifier methods combination with HOG features. The detailed measures of HOG and Naive Bayes combination is described in the table 3.1.37.

| Classifier/F-measures (%) | Balanced Dataset | Imbalanced Dataset |
|---------------------------|------------------|--------------------|
| Random Forest             | 61.0             | 43.1               |
| Adaboost                  | 56.7             | 20.0               |
| K-NN                      | 48.1             | 34.6               |
| Logistic regression       | 58.8             | 40.0               |
| RBF Network               | 63.9             | 24.6               |
| Naive Bayes               | <b>73.7</b>      | 57.8               |
| Decision Tree             | 59.5             | 22.2               |
| SVM                       | 62.5             | 0                  |

**Table 3.1.36 HOG features are extracted from balanced and imbalanced dataset, evaluated using 10-fold cross validation with different classifiers**

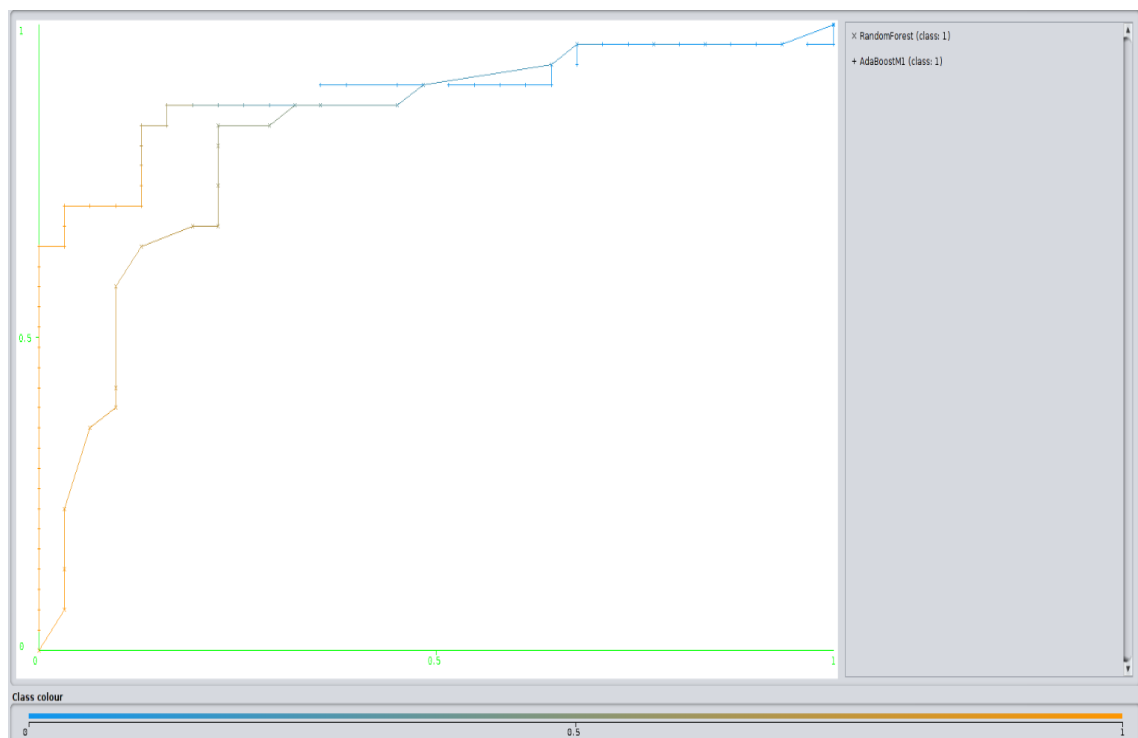
Even though the f-measure for class 1 is high, the overall accuracy and the accuracy metrics for class 0 is a bit low. Yet, the results are better than the Zernike accuracy metrics and slightly higher than the TAS and LBP features.



| Metrics(%)\Class | 0    | 1    | Weighted Avg. |
|------------------|------|------|---------------|
| TP Rate          | 45.2 | 90.3 | 67.7          |
| FP Rate          | 9.7  | 54.8 | 32.3          |
| Precision        | 82.4 | 62.2 | 72.3          |
| Recall           | 45.2 | 90.3 | 67.7          |
| F-Measure        | 58.3 | 73.7 | 66            |
| ROC Area         | 63.5 | 63.5 | 63.5          |
| Overall Accuracy |      |      | 67.7          |

**Table 3.1.37 HOG features are extracted from balanced dataset, evaluated using 10-fold cross validation with Naive Bayes classifier**

The ROC curves of the Adaboost and Random forest has been generated. Figure 3.1.1 describes the ROC curves. As the curve approximates to y-axis, the performance of the classifier is got better. For this reason, it can be said that AdaBoost performs better than the Random Forest.



**Figure 3.1.1 ROC curves of Random Forest and AdaBoost**

Summarization of the f-measure for balanced and normalized dataset with different classifiers and different features is showed on the table 3.1.38. Above other features Haralick gives stable and approximately higher results according to other classifiers.

| Classifier/Feature Descriptor (%) | LBP  | TAS  | ZERNIKE | GLCM        | HOG         | Haralick    |
|-----------------------------------|------|------|---------|-------------|-------------|-------------|
| Random Forest                     | 45.7 | 55.7 | 55.2    | 69.7        | 61.0        | <b>75</b>   |
| AdaBoost                          | 53.7 | 62.7 | 44.8    | 66.7        | 56.7        | <b>75</b>   |
| KNN                               | 40.0 | 59.7 | 56.3    | 63.5        | 48.1        | <b>73.8</b> |
| Logistic Regression               | 63.2 | 64.6 | 42.4    | <b>76.9</b> | 58.8        | <b>72.7</b> |
| RBF Network                       | 46.4 | 58.1 | 46.2    | <b>78.8</b> | 63.9        | 63.8        |
| Naïve Bayes                       | 47.2 | 66.7 | 55.1    | 70.0        | <b>73.7</b> | 61.7        |
| Decision Tree                     | 51.4 | 46.4 | 50.0    | 64.7        | 59.5        | 62.1        |
| SVM                               | 43.3 | 64.4 | 50.7    | <b>74.0</b> | 62.5        | <b>80</b>   |

**Table 3.1.38 Summarization of different classifiers and feature descriptors class 1's f-measure**

Feature selection algorithms are tested on Haralick's textural feature. Because, Haralick gives best accuracy metrics for both classes. Table 3.1.40 demonstrates the combination of different feature selection methods and different classifiers on balanced and imbalanced dataset with or without normalization. The values of the table represent the f-measure of the class 1. The feature selection algorithms are performed using their default parameter except Wrapper method. Wrapper methods uses classifier as a parameter. For each classification method Wrapper's classifier is set to current classification method. Best measures are written bold type.

Haralick's textural features give 56 features. Feature selection decreased the number of features. Each feature selection method has given different numbers: CFS selects 11 features. Correlation Attribute Evaluation, Gain Ratio, Info Gain, Relief Ranking and Symmetrical Uncertainty algorithms have ranked the features, first ten features of the ranked features have been selected. After PCA 8 new features have been produced by using 56 Haralick features. Wrapper have selected 6 features.

Random Forest classifier works well with, CFS and Wrapper feature selection method on balanced dataset. Normalization slightly changes the measure. Best performance of Random forest is achieved using Wrapper feature selector the f-measure for this combination is 83.9% for balanced and normalized dataset.

Logistic regression works well with correlation attribute evaluation on balanced dataset. For this case normalization does not affect the results. The f-measure is 84.4%.

for both normalized and unnormalized sets. Wrapper method also works well with Logistic regression. The f measure for balanced and unbalanced dataset is 81.8%.

KNN performs better when Wrapper is used on balanced dataset. The normalization does not change the result of the KNN classifier. The f-measure is 83.9%.

Naïve Bayes performs effectively with wrapper feature selection method on balanced dataset. Normalization does not change the metrics. Correlation attribute evaluation also works properly. The best f-measure that achieved is 81.3%.

Decision tree performs efficiently using Info Gain feature selector, this is followed by Gain Ration. Moreover, normalization gave approximately 5-6% better f-measure (81.3%) on balanced dataset.

SVM works well with wrapper method, normalization has been decreased the f-measure for this case. Without normalization the wrapper gives 81.3% f-measure while normalization gives 80%. On the other hand, without feature selection and normalization SVM gives 15.8%, for this case the normalization Boosts the SVMs performance and with normalization and without feature selection the f-measure is 80% on balanced dataset.

RBF Network give the best results when Wrapper feature selection method have been employed. The f-measure is 81.4% for abnormal class and this result is obtained on balanced dataset.

AdaBoost achieved highest score when CFS is employed on balanced dataset which is 84.8%. The normalization process does not change the f -measure.

| Combination of Classifier and Selector /       | Balanced    | Balanced and normalized | Imbalanced | Imbalanced normalized |
|--|-------------|-------------------------|------------|-----------------------|
| Random forest (without selection)              | 77.0        | 75.0                    | 79         | 80                    |
| CFS-Random forest                              | <b>80.0</b> | 78.8                    | 67.8       | 54.2                  |
| Correlation Attribute Evaluation-Random Forest | 75.8        | 73.8                    | 56.7       | 60.7                  |
| Gain Ratio-Random Forest                       | 77.4        | 76.2                    | 61.0       | 61.3                  |
| Info Gain - Random forest                      | 77.4        | 77.4                    | 57.1       | 59.6                  |
| PCA -Random forest                             | 75.6        | 78.9                    | 71.8       | 73.0                  |
| Relief Ranking - Random forest                 | 73.0        | 76.9                    | 68.9       | 71.0                  |

|   |             |             |      |      |
|---|-------------|-------------|------|------|
| Symmetrical<br>Uncertainty-<br>Random Forest                    | 75.4        | 79.4        | 61.0 | 62.1 |
| Wrapper- Random<br>Forest                                       | <b>82.5</b> | <b>83.9</b> | 79.4 | 80.0 |
| Logistic Regression<br>(without selection)                      | 72.7        | 72.7        | 73.8 | 75.0 |
| CFS-Logistic<br>Regression                                      | 76.2        | 76.2        | 71.9 | 71.9 |
| Correlation<br>Attribute<br>Evaluation –<br>Logistic Regression | <b>84.4</b> | <b>84.4</b> | 75.4 | 75.4 |
| Gain Ratio-<br>Logistic Regression                              | 80.6        | 80.6        | 58.6 | 58.6 |
| Info Gain –<br>Logistic Regression                              | 76.7        | 76.7        | 60.0 | 60.0 |
| PCA – Logistic<br>Regression                                    | 67.7        | 67.7        | 78.3 | 78.3 |
| Relief Ranking –<br>Logistic Regression                         | 79.4        | 79.4        | 72.1 | 72.1 |
| Symmetrical<br>Uncertainty-<br>Logistic Regression              | 76.7        | 76.7        | 53.3 | 53.3 |
| Wrapper – Logistic<br>Regression                                | <b>81.8</b> | <b>81.8</b> | 71.9 | 75.0 |
| KNN (without<br>selection)                                      | 73.8        | 73.8        | 63.5 | 63.5 |
| CFS - KNN   | 64.5        | 64.5        | 78.8 | 78.8 |
| Correlation<br>Attribute<br>Evaluation - KNN                    | 71.9        | 71.9        | 68.8 | 68.8 |
| Gain Ratio-KNN  | 71.0        | 67.7        | 60.3 | 60.3 |
| Info Gain - KNN   | 67.6        | 67.6        | 55.2 | 55.2 |
| PCA - KNN   | 75.7        | 75.7        | 71.3 | 71.3 |
| Relief Ranking -<br>KNN   | 68.7        | 68.7        | 71.0 | 71.0 |
| Symmetrical<br>Uncertainty- KNN                                 | 67.6        | 67.6        | 55.7 | 55.7 |
| Wrapper - KNN   | <b>83.9</b> | <b>83.9</b> | 76.2 | 76.2 |
| Naïve Bayes<br>(without selection)                              | 74.0        | 61.7        | 61.7 | 61.7 |
| CFS – Naïve Bayes   | 76.7        | 76.7        | 59.8 | 59.8 |
| Correlation<br>Attribute<br>Evaluation - Naïve<br>Bayes         | <b>80.0</b> | <b>78.8</b> | 68.3 | 68.3 |
| Gain Ratio - Naïve<br>Bayes                                     | 77.8        | 77.8        | 53.1 | 53.1 |
| Info Gain - Naïve<br>Bayes                                      | 77.8        | 76.7        | 53.1 | 53.1 |

|  |             |             |      |      |
|--|-------------|-------------|------|------|
| PCA - Naïve Bayes                                | 67.2        | 69.0        | 72.8 | 72.8 |
| Relief Ranking - Naïve Bayes                     | 79.4        | 81.3        | 68.5 | 68.5 |
| Symmetrical Uncertainty - Naïve Bayes            | 77.8        | 76.7        | 53.1 | 53.1 |
| Wrapper - Naïve Bayes                            | <b>81.3</b> | <b>81.3</b> | 73.7 | 75.4 |
| Decision Tree (without selection)                | 65.6        | 62.1        | 61.3 | 62.3 |
| CFS - Decision Tree                              | 72.1        | 79.4        | 63.0 | 60.7 |
| Correlation Attribute Evaluation - Decision Tree | 59.6        | 57.6        | 45.6 | 66.7 |
| Gain Ratio-Decision Tree                         | 74.2        | <b>80.0</b> | 65.3 | 68.0 |
| Info Gain - Decision Tree                        | 74.2        | <b>81.3</b> | 63.8 | 66.7 |
| PCA - Decision Tree                              | 61.3        | 61.3        | 70.1 | 70.1 |
| Relief Ranking - Decision Tree                   | 71.0        | 71.0        | 47.5 | 47.5 |
| Symmetrical Uncertainty - Decision Tree          | 74.2        | 81.3        | 65.3 | 68.0 |
| Wrapper - Decision Tree                          | 75.4        | 76.5        | 76.7 | 75.4 |
| SVM (without selection)                          | 15.8        | <b>80.0</b> | 0    | 63.0 |
| CFS - SVM  | 77.6        | 76.5        | 0    | 0    |
| Correlation Attribute Evaluation - SVM           | 72.1        | 72.1        | 51.1 | 47.8 |
| Gain Ratio - SVM                                 | 73.0        | 73.0        | 0    | 0    |
| Info Gain - SVM                                  | 73.0        | 73.0        | 0    | 0    |
| PCA - SVM  | 56.4        | 55.1        | 53.3 | 53.3 |
| Relief Ranking - SVM                             | 75.8        | 75.8        | 61.5 | 61.5 |
| Symmetrical Uncertainty - SVM                    | 73.0        | 73.0        | 0    | 0    |
| Wrapper - SVM                                    | <b>81.3</b> | <b>80.0</b> | 0    | 0    |
| RBF Network (without selection)                  | 63.8        | 63.8        | 63.8 | 57.6 |
| CFS - RBF Network                                | 72.5        | 72.5        | 70.4 | 70.4 |
| Correlation Attribute Evaluation - RBF Network   | 74.1        | 74.1        | 74.4 | 74.4 |

|   |             |             |      |      |
|---|-------------|-------------|------|------|
| Gain ratio - RBF Network                    | 75.8        | 75.8        | 77.3 | 77.3 |
| Info Gain - RBF Network                     | 77.4        | 77.4        | 74.7 | 74.7 |
| PCA - RBF Network                           | 72.5        | 72.5        | 79.3 | 79.3 |
| Relief Ranking - RBF Network                | 69.3        | 69.3        | 75.9 | 75.9 |
| Symmetrical Uncertainty - RBF Network       | 77.4        | 77.4        | 65.5 | 65.5 |
| Wrapper - RBF Network                       | <b>81.4</b> | <b>81.4</b> | 73.8 | 73.8 |
| AdaBoost (without selection)                | 75.0        | 75.0        | 64.5 | 64.5 |
| CFS - AdaBoost                              | <b>84.8</b> | <b>84.8</b> | 77.2 | 77.2 |
| Correlation Attribute Evaluation - AdaBoost | 76.7        | 76.7        | 42.3 | 67.6 |
| Gain Ratio-AdaBoost                         | 74.2        | 74.2        | 62.7 | 66.7 |
| Info Gain - AdaBoost                        | 78.8        | 78.8        | 67.9 | 66.7 |
| PCA - AdaBoost                              | 71.0        | 71.0        | 75.4 | 75.4 |
| Relief Ranking - AdaBoost                   | 70.8        | 70.8        | 55.2 | 63.5 |
| Symmetrical Uncertainty-AdaBoost            | 78.8        | 78.8        | 62.7 | 69.3 |
| Wrapper - AdaBoost                          | 79.4        | 79.4        | 73.8 | 69.1 |

**Table 3.1.39 Feature selection methods evaluated using different classifiers**

Parameter optimization of KNN, RBF Network, Adaboost, SVM and Random Forest is performed on balanced Haralick features dataset whose features were selected according to classifier.

The irrelevant features of the dataset were eliminated using CFS feature selection. After, parameter of the KNN (K) is optimized using the following numbers 1, 5, 10, 15, ... 100 (with increments of 5 after k=5). KNN gave the best result when k=20. Table 3.1.41 shows the detailed metrics.

| Metrics(%<br>)\Class | 0    | 1           | Weighted<br>Avg. |
|----------------------|------|-------------|------------------|
| TP Rate              | 71   | 90.3        | 80.6             |
| FP Rate              | 9.7  | 29          | 19.4             |
| Precision            | 88   | 75.7        | 81.8             |
| Recall               | 71   | 90.3        | 80.6             |
| F-Measure            | 78.6 | <b>82.4</b> | 80.5             |
| ROC Area             | 81.7 | 81.7        | 81.7             |
| Overall<br>Accuracy  |      |             | <b>75</b>        |

**Table 3.1.40 Accuracy metrics of KNN when k=20**

The irrelevant features of the dataset were eliminated using CFS feature selection. Then, parameter of the Random Forests (number of trees) optimized using the following numbers 5, 10, 25, 50, 75, 100, 150, 200, 250, 300, 350, 400, 450, 500. The optimum number of tree was 25. Table 3.1.42 shows the detailed metrics.

| Metrics(%<br>)\Class | 0    | 1           | Weighted<br>Avg. |
|----------------------|------|-------------|------------------|
| TP Rate              | 77.4 | 83.9        | 80.6             |
| FP Rate              | 16.1 | 22.6        | 19.4             |
| Precision            | 82.8 | 78.8        | 80.8             |
| Recall               | 77.4 | 83.9        | 80.6             |
| F-Measure            | 80   | <b>81.3</b> | 80.6             |
| ROC Area             | 83.4 | 83.4        | 83.4             |
| Overall<br>Accuracy  |      |             | <b>80.64</b>     |

**Table 3.1.41 Accuracy metrics of Random Forest when number of trees equal to 25**

Number of clusters for RBF Network has been optimized using the parameter set of 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30, 35, 40, 45, 50. Best result was achieved when number of clusters are equal to 15. Table 3.1.43 shows the accuracy metrics.

| Metrics(% )\Class | 0    | 1           | Weighted Avg. |
|-------------------|------|-------------|---------------|
| TP Rate           | 67.7 | 77.4        | 72.6          |
| FP Rate           | 22.6 | 32.3        | 27.4          |
| Precision         | 75   | 70.6        | 72.8          |
| Recall            | 67.7 | 77.4        | 72.6          |
| F-Measure         | 71.2 | <b>73.8</b> | 72.5          |
| ROC Area          | 76.5 | 76.3        | 76.4          |
| Overall Accuracy  |      |             | <b>70.64</b>  |

**Table 3.1.42 Accuracy metrics of RBF Network when number of clusters equal to 15**

Number of iterations for Adaboost classifier has been changed using the following set of numbers: 5, 10, 15, 20, 25, 30, 40, 50, 75, 100, 125, 150, 175, 200. Best result was performed when number of iterations are equal to 50. Table 3.1.44 shows the accuracy metrics.



| Metrics(%<br>)\Class | 0    | 1    | Weighted<br>Avg. |
|----------------------|------|------|------------------|
| TP Rate              | 83.9 | 87.1 | 85.5             |
| FP Rate              | 12.9 | 16.1 | 14.5             |
| Precision            | 86.7 | 84.4 | 85.5             |
| Recall               | 83.9 | 87.1 | 85.5             |
| F-Measure            | 85.2 | 85.7 | <b>85.5</b>      |
| ROC Area             | 89   | 89   | 89               |
| Overall<br>Accuracy  |      |      | <b>85.48</b>     |

**Table 3.1.43 Accuracy metrics of AdaBoost when number of iterations equal to 50**

To optimize the C and gamma parameters of SVM the following parameter sets has been used.

$$C \in \{2^{-5}, 2^{-3}, 2^{-1}, 2^1, 2^3, 2^5, 2^7, 2^{10}, 2^{13}, 2^{15}\}$$

$$\gamma \in \{2^{-15}, 2^{-13}, \dots, 2^{-1}, 2^1, 2^3, 2^5\}$$

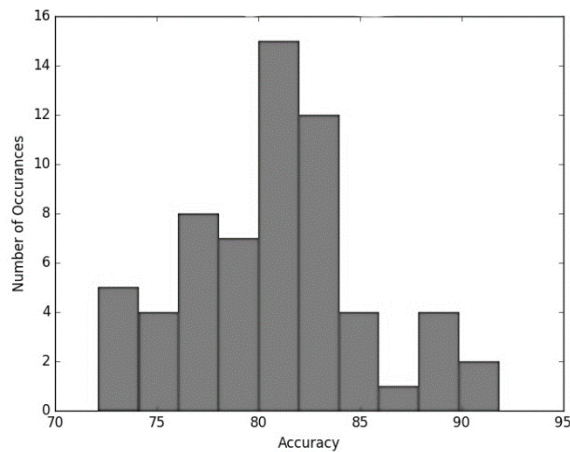
There were total 11 values for the C parameter and 11 parameters for gamma parameter. A total of 121 pair has been performed. Among all the pairs, the combination of  $C = 2^{13}$  and  $\gamma = 2^{-5}$  gives the best result. Table 3.1.45 shows the detailed metrics.

| Metrics(%<br>)\Class | 0    | 1           | Weighted<br>Avg. |
|----------------------|------|-------------|------------------|
| TP Rate              | 74.2 | 90.3        | 82.3             |
| FP Rate              | 9.7  | 25.8        | 17.7             |
| Precision            | 88.5 | 77.8        | 83.1             |
| Recall               | 74.2 | 90.3        | 82.3             |
| F-Measure            | 80.7 | <b>83.6</b> | 82.1             |
| ROC Area             | 82.3 | 82.3        | 82.3             |
| Overall<br>Accuracy  |      |             | <b>82</b>        |

**Table 3.1.44 Accuracy metrics of SVM when  $C = 2^{13}$  and  $\gamma = 2^{-5}$**

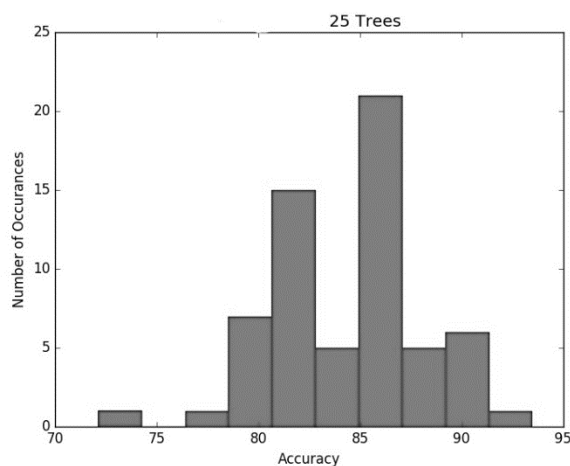
The optimum parameters graph of occurrence vs accuracy has been drawn for SVM, AdaBoost and Random Forest since their optimum parameters were achieved the best results of optimization.

Figure 3.1.2 shows AdaBoost with 50 iterations, accuracy versus number of occurrences of the accuracy. The number of occurrences has been concentrated on 80-85%.



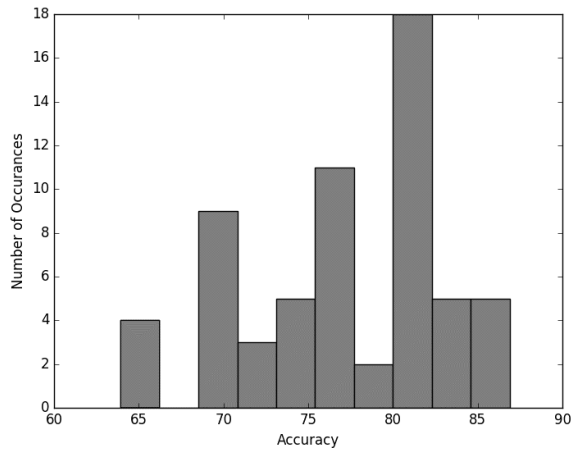
**Figure 3.1.2 AdaBoost occurrence of the accuracy versus number of iterations for 50 iterations**

Figure 3.1.3 shows Random Forest with 25 trees (iterations), accuracy versus number of occurrences of the accuracy. The number of occurrences has been concentrated on 80-90%.



**Figure 3.1.3 Random Forest occurrence of the accuracy versus number of trees for 25 trees**

SVM with parameters  $C = 2^{13}$  and  $\gamma = 2^{-5}$  graph is shown in the figure 3.1.3. The number of occurrences has been concentrated on the range between 75-85%.



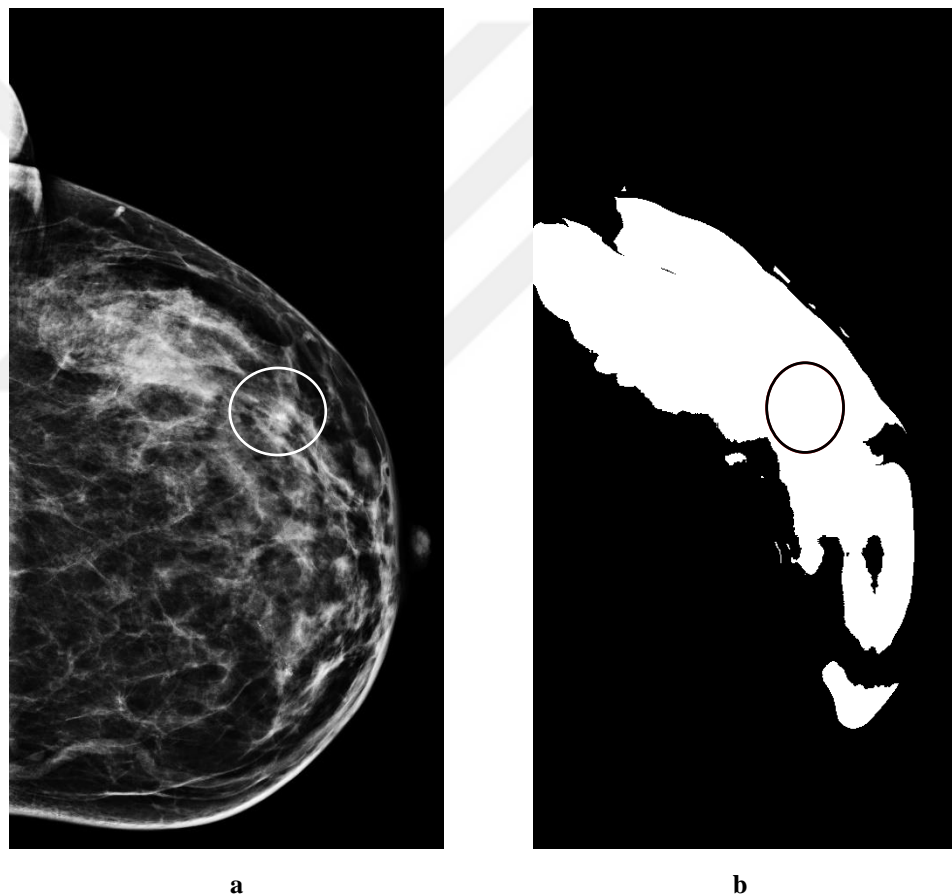
**Figure 3.1.4 SVM occurrence of the accuracy versus different parameter for  $C = 2^{13}$  and  $\gamma = 2^{-5}$**

Leave-one-instance-out cross validation (LOO-CV) has been employed for best classifier methods on balanced Haralick feature set whose features are selected using Wrapper methods. Table 3.1.46 shows overall accuracy metrics of classifiers' evaluation using LOO-CV. One can understand that there was a little bit overfitting because the metrics has been decreased after LOO-CV. The metrics of Random Forest and AdaBoost are approximately close to each other. On the other hand, Adaboost and SVM metrics have been decreased dramatically because they are more prone to overfitting while Random Forest more robust to overfitting. Furthermore, the overall accuracies for 10-fold CV of Random Forest, SVM, AdaBoost are 80.64, 82, 85.48.

| Metrics(%)<br>/Classifier | Random<br>Forest | SVM   | AdaBoost    |
|---------------------------|------------------|-------|-------------|
| Accuracy                  | 77.5             | 70.96 | 77.4        |
| F-measure                 | 75.9             | 70.2  | <b>77.3</b> |
| FP-Rate                   | 18.8             | 29    | 22.6        |
| Specificity               | 81.3             | 71.0  | 77.4        |
| Precision                 | 78.6             | 73.4  | 77.9        |
| Recall                    | 73.3             | 71    | <b>77.4</b> |

**Table 3.1.45 LOO-CV performed on best classifiers using Haralick features**

Random Forest and Haralick combination constructed a new model. Using this model, mammograms has been traversed window by window and each window has been labeled as abnormal or normal. The zero and one labels have been represented as black and white color respectively. Afterwards, new mammograms have been reconstructed using the labels of the windows. The purpose of this process is to find the ROI of a mammogram by examining the entire mammogram. Figure 3.1.4 shows a mammogram and its reconstructed version using model. It can be seen that the model labeled some normal regions as abnormal and it labelled the abnormal area as abnormal. There are excess false alarms. For this reason, the model should be developed.



**Figure 3.1.5 The mammogram (a) is traversed window by window and labeled using the model, labels reconstruct the mammogram (b)**

Stacking method has been employed on balanced dataset with the classifiers that has gave the well results on the previous experiments. Meta classifier of the stacking has been selected as Logistic Regression Classifier. The reason of this selection is Logistic

Regression Classifier is simple, using the complicated meta learner decreases the accuracy.

Table 3.1.47 describes the f-measure for stacking of AdaBoost, RBF Network using different meta learner on balanced dataset. Moreover, the table shows KNN, AdaBoost combination and AdaBoost, Random Forest combination with meta learner of Logistic regression on balanced dataset. Also, SVM classifier is used as meta learner and the performance is compared with Logistics performance. As it shown the table 3.1.47 the f-measure has been decreased approximately 9% when the SVM is used as meta learner.

Furthermore, combination of Adaboost and RBF Network with Logistic meta learner achieves best accuracy among others.

| Combinations( %)\Class  | 0    | 1           | Weighted Avg. |
|---|------|-------------|---------------|
| Adaboost and RBF Network with meta learner Logistic Regression            | 83.3 | <b>84.4</b> | 83.9          |
| Adaboost and RBF Network combination with meta learner SVM                | 77.4 | 74.2        | 75.8          |
| KNN, AdaBoost combination with meta learner Logistic Regression           | 76.7 | 78.1        | 77.4          |
| AdaBoost, Random Forest combination with meta learner Logistic Regression | 80   | <b>81.3</b> | 80.6          |

**Table 3.1.46 Different combination of classifiers has been performed using Logistic and SVM meta learner**

Combination of different feature extraction methods have been performed and Random Forest is employed as classifier with 10-fold CV. Random forest is used because it is more robust to overfitting. Table 3.1.48 shows the results. The figure shows f-measure of class 1's with and without feature selection. The results are normalized. Feature selection is done by using Wrapper method. Wrapper is selected, the reason for this selection is according to 3.1.48 most of the cases Wrapper gives best or second-best

results. HOG, Haralick combination and Haralick, TAS combination gives best accuracy among other combination and previous feature extraction methods. This is followed by, Tas, HOG combination and GLCM, Haralick combination. For all the cases, feature selection increases the performance of the method.

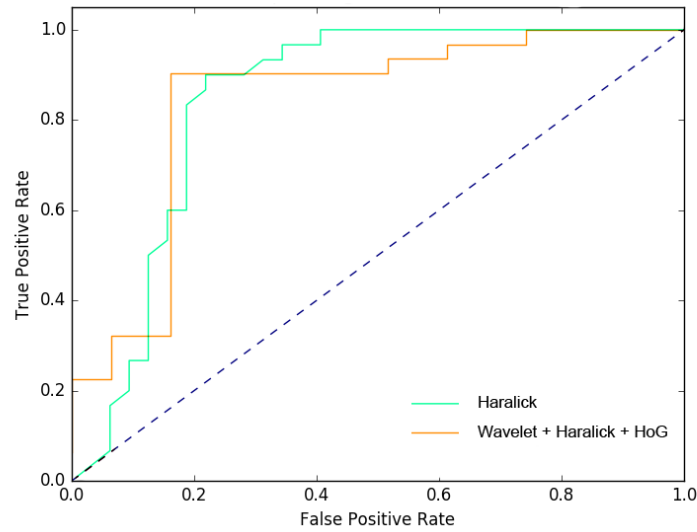
| Combination of Feature extraction/F-measures (%) | Without feature selection | With feature selection |
|--|---------------------------|------------------------|
| GLCM and Haralick                                | <b>76.2</b>               | <b>76.9</b>            |
| GLCM and HOG                                     | 58.6                      | 71.9                   |
| GLCM and TAS                                     | 50.9                      | 67.7                   |
| Haralick and TAS                                 | 76.2                      | <b>80.6</b>            |
| HOG and Haralick                                 | 75.0                      | <b>80.6</b>            |
| TAS and HOG                                      | 44.8                      | <b>78.5</b>            |

**Table 3.1.47 Different combination of features are extracted from balanced dataset, feature selection is made by Wrapper method, then selected and non-selected versions is evaluated using 10-fold cross validation with Random Forest classifier using default parameters**

Some experiments of feature extraction algorithms have been repeated using Wrapper feature selector and LOO-CV algorithm. Moreover, new datasets have been constructed using Wavelet decomposition employed as new component of feature extraction algorithms. Hybrid of wavelet and other feature extraction algorithms is used. To extract the hybrid features first wavelet decomposition is applied and then the features are extracted from LL component of the wavelet decomposition. Then extracted features are normalized using min-max normalization. Irrelevant features are eliminated using wrapper feature selector. Random Forest classifier have been employed, number of trees has been optimized for each dataset. The accuracy measures are represented on table 3.1.49. It can be seen from the table that combination of Wavelet, Haralick and HOG descriptors outperforms which means this combination is useful for feature selection. The figure 3.1.5 shows the ROC curve of the proposed algorithm.

| Method / Measure                           | Accuracy     | F-Measure    | FP Rate      | Specificity  | Precision    | Recall       | AUC Score  |
|--|--------------|--------------|--------------|--------------|--------------|--------------|------------|
| Haralick                                   | <b>77.5%</b> | <b>75.9%</b> | <b>18.8%</b> | <b>81.3%</b> | <b>78.6%</b> | <b>73.3%</b> | <b>85%</b> |
| HOG  | 43.5%        | 42.6%        | 56.3%        | 43.8%        | 41.9%        | 43.3%        | 44%        |
| LBP  | 50%          | 45.6%        | 43.7%        | 56.3%        | 48.1%        | 43.3%        | 46%        |
| TAS  | 56.45%       | 55.7%        | 43.8%        | 56.3%        | 54.8%        | 56.7%        | 62%        |
| Wavelet + Haralick                         | 71%          | 70%          | 28.1%        | 71.9%        | 70%          | 70%          | 79%        |
| Wavelet + HOG                              | 46.8%        | 50.7%        | 62.5%        | 37.5%        | 45.9%        | 56.6%        | 44%        |
| Wavelet + GLCM                             | 58.1%        | 58.1%        | 43.8%        | 56.3%        | 56.3%        | 60%          | 66%        |
| Wavelet + Haralick + HOG (Proposed method) | <b>87.1%</b> | <b>87.5%</b> | <b>16.1%</b> | <b>83.8%</b> | <b>84.8%</b> | <b>90%</b>   | <b>84%</b> |

**Table 3.1.48 LOO-CV performed on different feature extraction algorithms using Random forest algorithm.**



**Figure 3.1.6 ROC curves of the Haralick and the proposed Wavelet-Haralick-HOG descriptor (feature selection is applied in LOOCV setting for each method)**

Wavelet, Haralick and HOG features are extracted from ROIs. The features are normalized and selected using Wrapper feature selector. Then, proposed method was evaluated using Random Forest, SVM, AdaBoost classification algorithms. Table 3.1.50 shows the LOO-CV results of the proposed method. One can understand that the proposed method performs successfully using Random Forest classifier. It is followed by SVM classifier. Adaboost performs effectively on the Haralick dataset and 10-fold CV, while classifies poorly when LOO-CV and proposed method is adopted.

| Classifier / Measure | RF            | SVM | AdaBoost |
|----------------------|---------------|-----|----------|
| Accuracy             | <b>87.10%</b> | 77% | 69%      |
| F-Measure            | <b>87.50%</b> | 78% | 69%      |
| FP Rate              | 16.10%        | 25% | 32%      |
| Specificity          | 83.80%        | 74% | 67%      |
| Precision            | 84.80%        | 75% | 68%      |
| Recall               | 90%           | 80% | 71%      |
| AUC Score            | 84%           | 77% | 72%      |

**Table 3.1.49 LOO-CV performed on proposed feature extraction algorithm using different classifiers**

The experiments on the Pilot dataset showed that using balanced dataset gives more accurate results. Furthermore, some classifiers are prone to overfitting in order to



have true results (without overfitting) LOO-CV can be performed and the true results can be obtained. Combination of true feature extraction methods, feature selection and classifier give higher accuracy results.

### 3.2 Image enhancement on mammographic images

Some image enhancement techniques are applied in order to emphasize the tumorous region on MIAS dataset. In Pilot dataset the quality of the images is better therefore the malignant regions are more visible. Figure 3.2.1 a and b shows samples from both datasets.

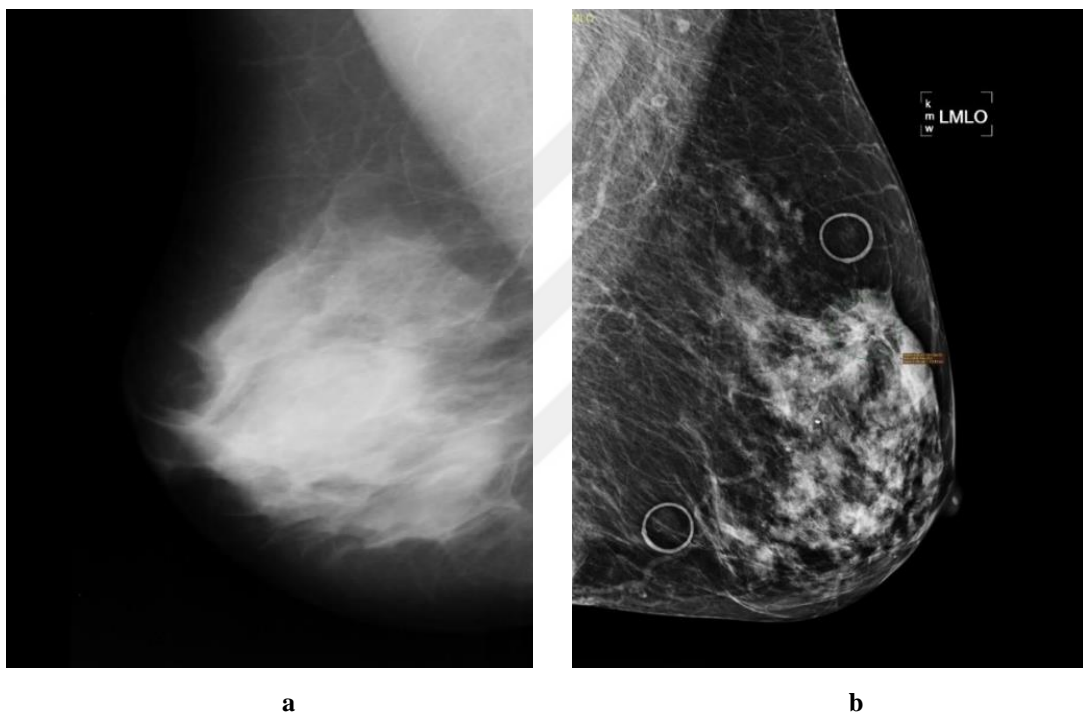
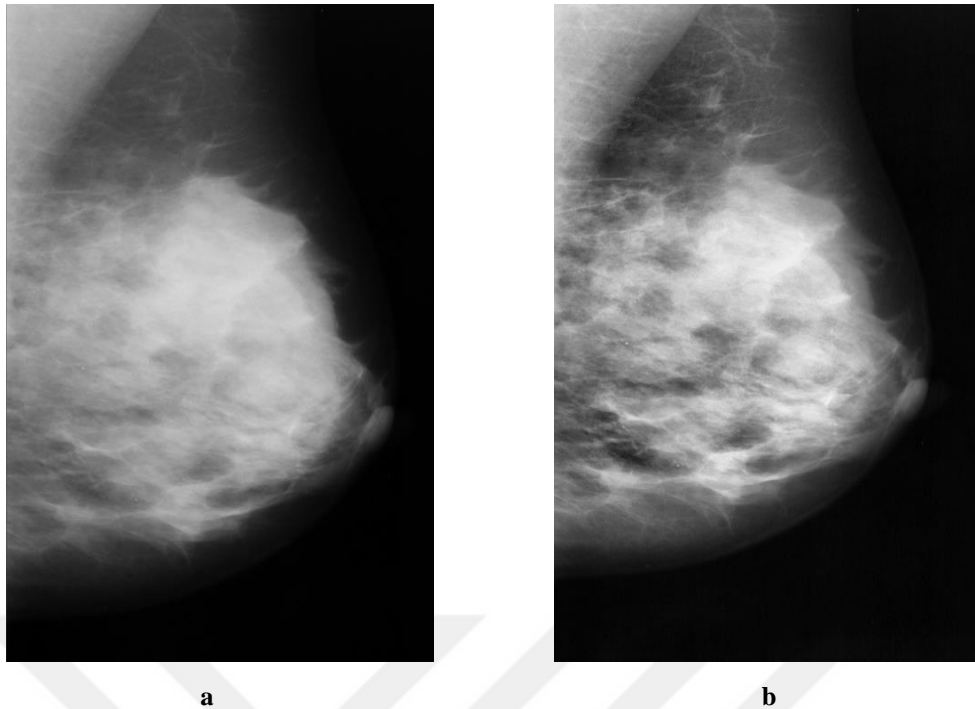
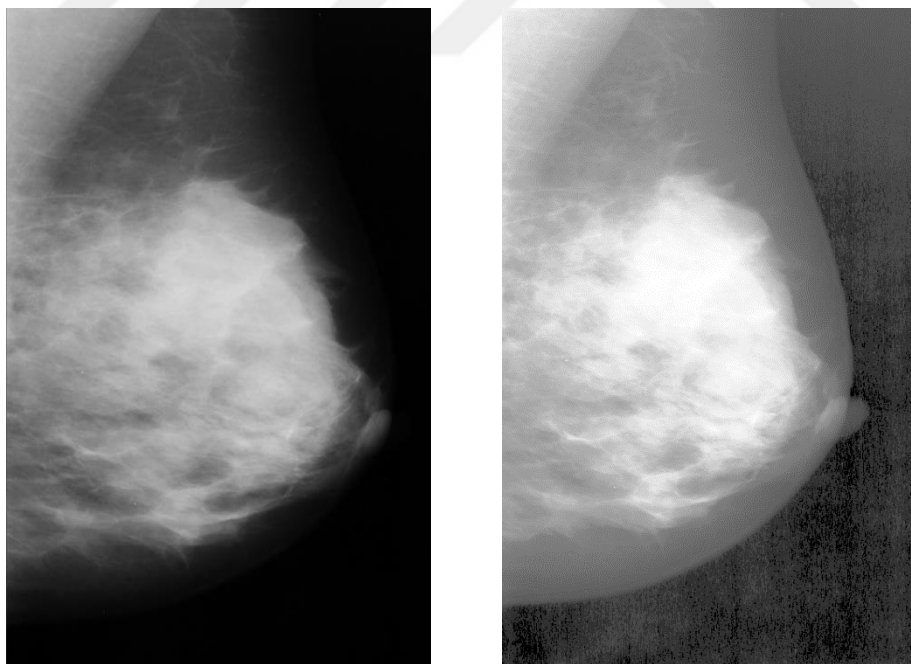


Figure 3.2.1 a is a sample from MIAS dataset and b is from pilot dataset.

CLAHE operation is applied in order to increase the visibility of the tumorous region in MIAS dataset. Figure 3.2.2 shows the result of CLAHE processed version (b) of the original image (a).



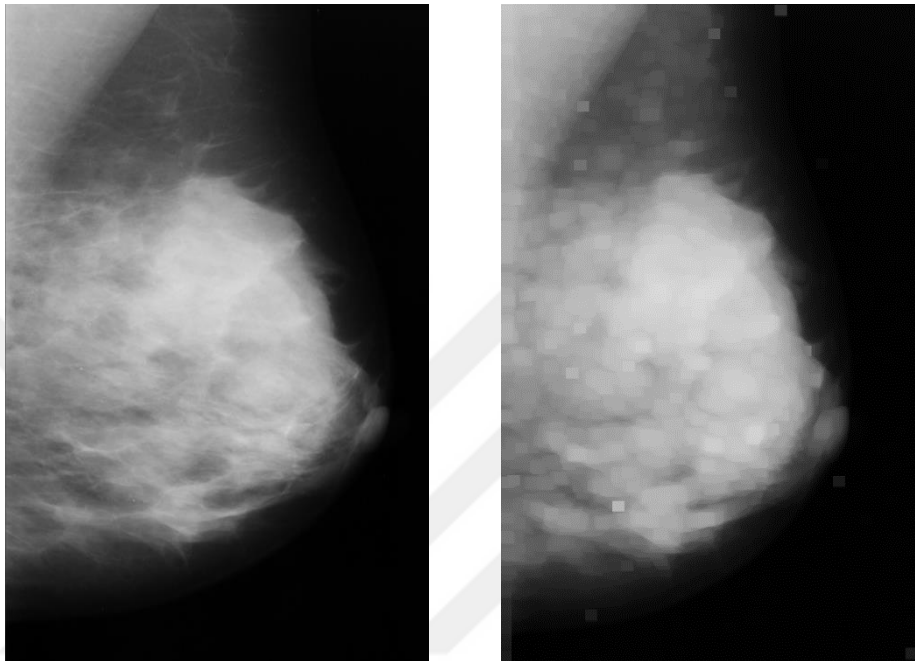
**Figure 3.2.2 a) original mammogram and B) CLAHE processed version of original**  
 Histogram equalization is applied to adjust the contrast of the image. Figure 3.2.3 shows histogram equalized version (b) of the original image (a)



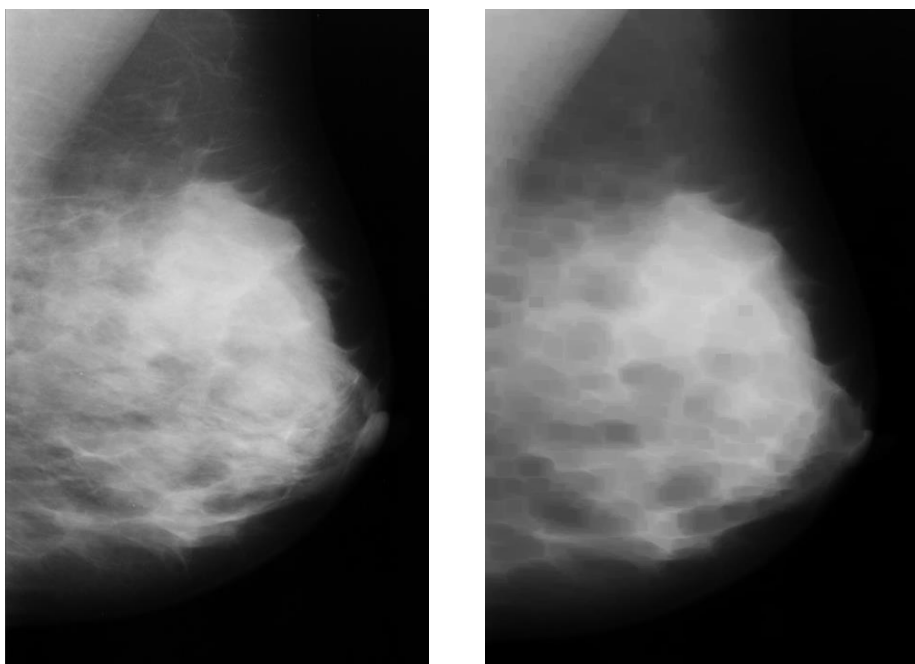
**Figure 3.2.3 a) original image b) histogram equalized version of the original image**

Some basic image processing operations has been applied to highlight the suspicious areas, figure 3.2.4, figure 3.2.5, figure 3.2.6, figure 3.2.7, figure 3.2.8, figure 3.2.9 shows original mammogram and dilation, erosion, opening, closing, white top hat,

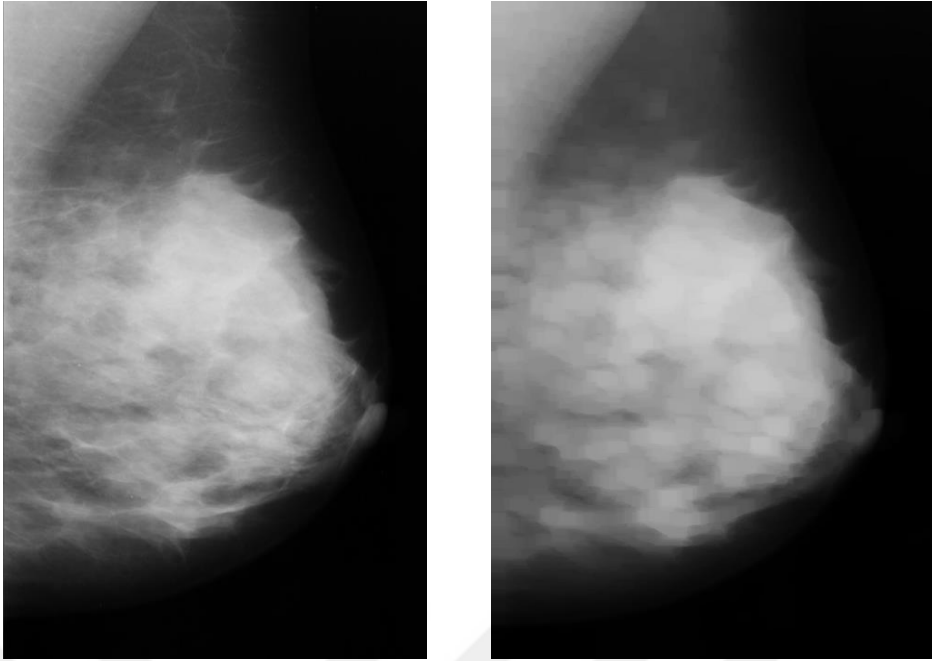
black top hat processed version of the original mammogram image, respectively. The structuring element that is used for dilation, erosion, opening and closing process is 15x15 square. White top-hat and black top-hat operations structuring element is 250x250 square. One can interpret by visual inspection that among all image processing operations, CLAHE gave the best result.



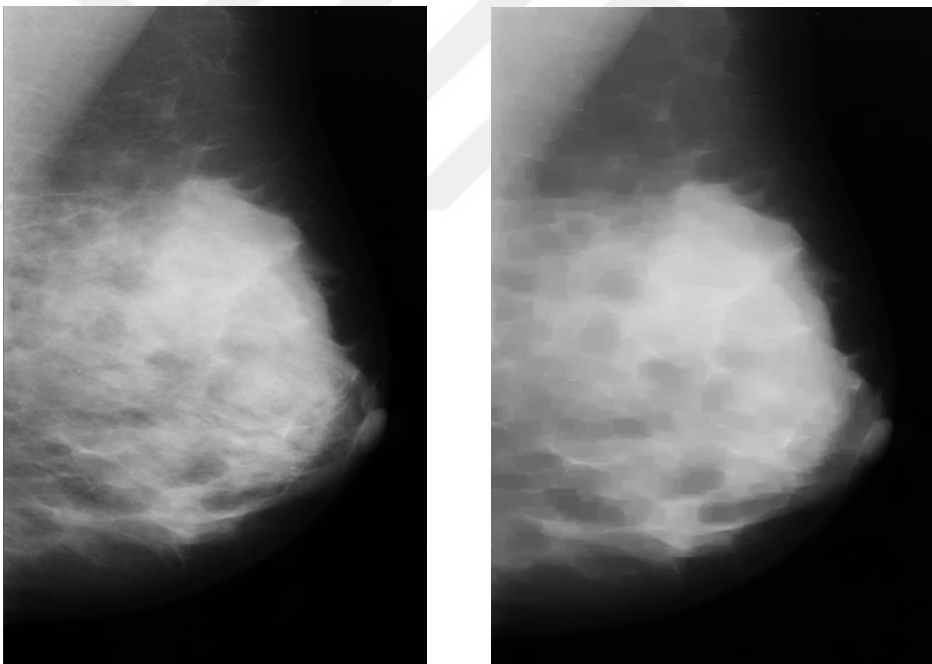
**Figure 3.2.4 Original mammogram and Dilation applied mammogram**



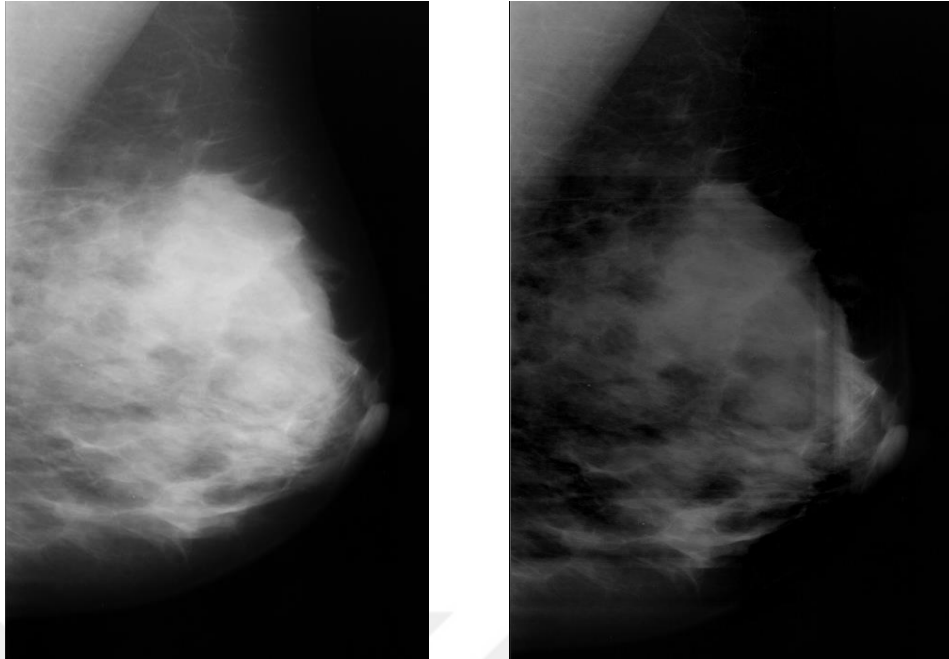
**Figure 3.2.5 Original mammogram and Erosion applied mammogram**



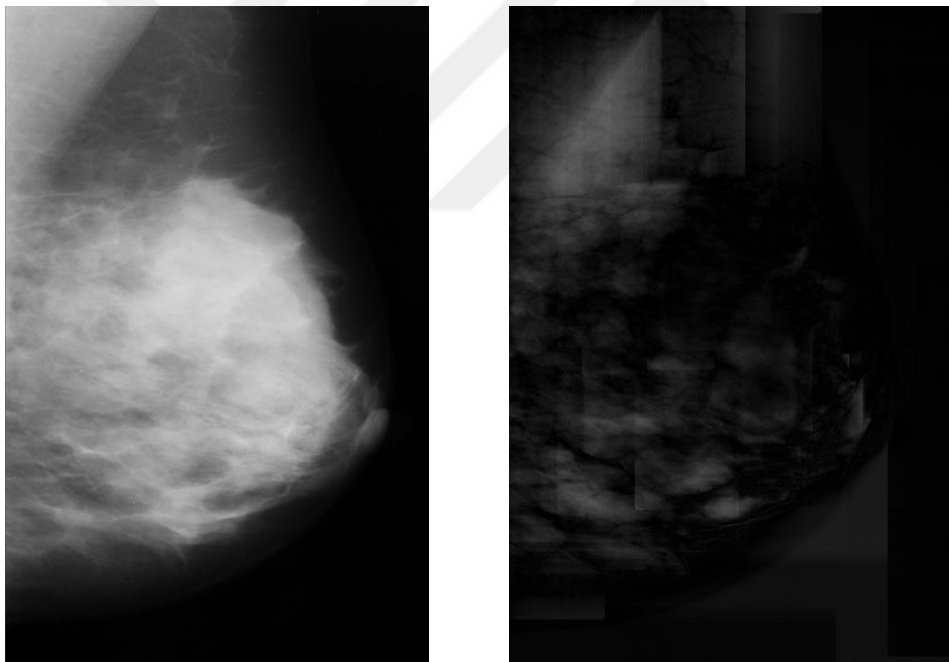
**Figure 3.2.6 Original mammogram and Opening applied mammogram**



**Figure 3.2.7 Original mammogram and Closing applied mammogram**



**Figure 3.2.8 Original mammogram and Top-Hat (white top-hat) applied mammogram**



**Figure 3.2.9 Original mammogram and Black Top-Hat applied mammogram**

CLAHE is selected and used for all MIAS dataset. Table 3.2.1 shows 10-fold cross validation of extracted features using proposed method from original MIAS dataset while table 3.2.2 shows CLAHE applied results. It can be seen that the CLAHE applied version has been increased the results by 6%.

| Metrics(%)\Class | 0    | 1         | Weighted Avg. |
|------------------|------|-----------|---------------|
| TP Rate          | 66.5 | 30.6      | 52            |
| FP Rate          | 69.4 | 33.5      | 54.9          |
| Precision        | 58.6 | 38.2      | 50.4          |
| Recall           | 66.5 | 30.6      | 52            |
| F-Measure        | 62.3 | <b>34</b> | 50.9          |
| ROC Area         | 54.2 | 54.2      | 54.2          |
| Overall Accuracy |      |           | 52            |

**Table 3.2.1. Proposed methods features are extracted from MIAS database and evaluated using Random Forest classifier with 10-fold CV**

| Metrics(%)\Class | 0    | 1           | Weighted Avg. |
|------------------|------|-------------|---------------|
| TP Rate          | 73.6 | 36.9        | 58.8          |
| FP Rate          | 63.1 | 26.4        | 48.2          |
| Precision        | 63.2 | 48.8        | 57.3          |
| Recall           | 73.6 | 36.9        | 58.8          |
| F-Measure        | 68   | <b>42.1</b> | 57.5          |
| ROC Area         | 65.2 | 65.2        | 65.2          |
| Overall Accuracy |      |             | 58.8          |

**Table 3.2.2 Proposed methods features are extracted from CLAHE applied MIAS database and evaluated using Random Forest classifier with 10-fold CV**

Multi-layer perceptron (MLP) has been applied to MIAS ROI database whose features are extracted using the proposed method. MLP had one hidden layer and number of hidden units were 25, the activation function of the hidden neurons and output neuron was hyperbolic tangent, cost function was cross entropy. The optimization algorithm was the steepest gradient descent algorithm (i.e. online mode) using back-propagation.

Weights are initialized randomly using uniform distribution in range  $[-1,1]$ . A fixed learning rate  $\eta = 0.1$  have been used in the beginning. Stopping method has been implemented using the early stopping criteria which can be implemented using the following steps: 10% of the dataset used as train set and another 10% is used as validation set. Maximum of 20 epochs has been performed and optimum number of epoch and iterations has been found. Once the optimum number of iterations/epochs was found the network was trained using the rest of the dataset. The outputs have been predicted using the procedure above with and without feature selection. As feature selection wrapper method has been employed. Moreover, prediction part is repeated 10 times. Different number of optimum epoch/iteration and accuracy rates were obtained. Table 3.2.3 and 3.2.4 shows the results, feature selection does not change for one evaluation and normalization increased the accuracy. When the evaluation was repeated 10 times (10-fold) feature selection achieves better results. For one-fold optimum number of epoch and iteration was 1 epoch and zero iteration while, the optimum parameters for ten-fold was 4 epoch and 2 iteration for normalized dataset.

|                           | Without Feature Selection(Normalized) | With Feature Selection (Normalized) |
|---------------------------|---------------------------------------|-------------------------------------|
| Accuracy                  | 75%                                   | 75%                                 |
| Mean accuracy for 10-fold | 70.35% $\pm$ 7.59%                    | 72.5% $\pm$ 2.78%                   |

**Table 3.2.3 MLP accuracy result of one-fold and ten-folds on normalized dataset**

|                           | Without Feature Selection(without Normalization) | With Feature Selection (without Normalization) |
|---------------------------|--|--|
| Accuracy                  | 57%  | 67%  |
| Mean accuracy for 10-fold | 69% $\pm$ 7.53%                                  | 72.85% $\pm$ 2.36%                             |

**Table 3.2.4 MLP accuracy result of one-fold and ten-folds on dataset**

Different learning rates have been performed using the search-then-converge schedule using equation 3.2.1, where  $n$  is number of iterations (1, ... 100.000),  $\eta_0 = 0.1$  and  $\tau = 10^3$ . The accuracy of 64.28% have been obtained on normalized dataset while,

the search-then-converge was performed on dataset whose features nor selected nor normalized, the accuracy of **71.6%** have been achieved.

$$\eta(n) = \frac{\eta_0}{1 + \frac{n}{\tau}} \quad (\text{Equation 3.2.1})$$

Momentum coefficient  $\alpha$  was included to weight update rule, for this experiment learning rate have been fixed to 0.1 again, accuracy of 64.42 has been obtained. On the other hand, the experiment is repeated on normalized dataset whose features were selected **71.42%** accuracy have been achieved.

MLP that have been described previously is repeated using Batch optimization. Feature selection and normalization have been performed on the data set and the experiments were done on this dataset. The accuracy of **75%** have been achieved for one-fold MLP, however for ten-fold MLP the accuracy decreased critically to **55%±24%**. The accuracy of **55%** and **75%** has been obtained for search then converge and momentum coefficients respectively. Most of the cases the performance of the online learning is higher than the Batch learning.

Mini Batch mode has been employed and MLP is repeated using mini batch. The normalized dataset whose features have been selected was used in experiments. One folds accuracy was **75%** while ten-fold accuracy was decreased 1% compared with the Batch mode and became **54%±24%**. The momentum coefficients and the momentum coefficients gave the same result of Batch learning.

The parameters of mini Batch have been optimized. The optimized parameters were: number of epochs ( $N_e$ ), number of hidden units at the hidden layer (M), learning rate ( $\eta$ ) and the mini Batch size ( $B_s$ ) and momentum coefficient ( $\alpha$ ) using the range:

$$1 \leq N_e \leq 20$$

$$M = 50, 100, 150, 200, 250$$

$$B_s = 25, 50, 75, 100$$

$$\eta = 0.5, 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005$$

$$\alpha = 0.5, 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005$$

The highest accuracy of **75%** is achieved when  $N_e = 1$ ,  $M=25$ ,  $\alpha=0.5$  and  $\eta = 0.01$ . The optimization gave the same results.



The same optimization previously that was performed, has been repeated and this time instead of optimizing the momentum coefficient, search-then-converge schedule has been optimized using the following values of  $\eta_0$ :

$$\eta_0 = 0.5, 0.1, 0.05, 0.01, 0.005$$

The optimum accuracy **75%** was achieved when the  $N_e = 8$ ,  $M=20$  and  $\eta_0 = 0.1$ . The time of the convergence was equal to previous experiments time of convergences. The accuracy stayed the same.

L2-norm regularization have been performed with optimization of the lambda ( $\lambda$ ) and other parameters that have been optimized on pervious experiments. The range of the lambda values were [0.0001, 0.001, 0.01, 0.1, 1.0, 10,100]. The max accuracy that achieved in this experiment was **70%** when  $M=5$ ,  $\lambda=0.001$ ,  $B_s=50$ ,  $\eta_0=0.1$ .

Some optimization techniques (ADAM, Adagrad, Adadelata and Rprop) have been applied in order to improve the MLP algorithm results are shown in table 3.2.5. Adagrad achieved the **70.3%** accuracy when initial accumulator was 0.01. Initial accumulators default value is 0.1 which is used by TensorFlow. Using Adadelata **57%** accuracy have been obtained when ADAM decay rate was equal to 0.8, default value of 0.95 is used by TensorFlow. ADAM and RMSProp achieve **81.4%** and **74%** accuracy respectively. RMSProp's decay rate was 0.5 (default value: 0.9), ADAM's beta1 was 0.7 (default: 0.9) and beta2 was 0.99 (default: 0.99).

| Parameters/Optimization Technique | Accuracy     | Number of Hidden Neurons | Learning Rate | Minibatch Size |
|-----------------------------------|--------------|--------------------------|---------------|----------------|
| Adagrad                           | 70.3%        | 15                       | 0.001         | 25             |
| Adadelata                         | 57%          | 30                       | 0.5           | 75             |
| ADAM                              | <b>81.4%</b> | 25                       | 0.0005        | 25             |
| RMSProp                           | 74%          | 30                       | 0.001         | 50             |

**Table 3.2.5 MLP accuracy results and optimum parameters for different optimization techniques.**

MLP with two and three hidden layers is used. The construction of this new MLPs was similar to previously explained MLP except hidden layer number. The search then converge schedule have been applied. Table 3.2.6 describes the accuracy results and optimum parameters for 2 and 3 hidden layers MLP. One can understand that the increasing the number of hidden layers does not increases the accuracy of the system.

| Parameters/Optimization Technique | Accuracy | Number of Hidden Neurons | $\eta_0$ | Minibatch Size |
|-----------------------------------|----------|--------------------------|----------|----------------|
| 2 Hidden Layer                    | 74%      | 5                        | 0.001    | 25             |
| 3 Hidden Layer                    | 74%      | 5                        | 0.005    | 75             |

**Table 3.2.6 MLP accuracy results and optimum parameters for 2 and 2 hidden layer MLP**

The 2 and 3 hidden layer MLP have been optimized using the ADAM optimization the **81.4%** accuracy have been achieved for both cases.

Table 3.2.7 summarizes the accuracy results for different experiments that have been already done. The best result has been achieved when the ADAM optimization was performed.

| Experiments   | Accuracy     |
|---|--------------|
| Optimization of Parameters using different learning rates | 75%          |
| Optimization using Search then converge                   | 75%          |
| L2-norm   | 70%          |
| Adagrad   | 70.3%        |
| Adadelta  | 57%          |
| RMSProp   | 74%          |
| <b>ADAM</b>   | <b>81.4%</b> |
| 2 hidden layer MLP  | 74%          |
| 3 hidden layer MLP  | 74%          |
| <b>ADAM and 2 hidden layer MLP</b>                        | <b>81.4%</b> |
| <b>ADAM and 3 hidden layer MLP</b>                        | <b>81.4%</b> |

**Table 3.2.7 MLP accuracy results of previous experiments**

Basic machine learning classifiers have been performed to evaluate the performance of the proposed method on MIAS database without feature selection. Table 3.2.8 describes the 10-fold accuracy of different classifiers using default parameters. Best accuracy measures have been obtained when Bagging and Random Forest classifiers was employed.

| Classifier/Accuracy  | Accuracy (%) |
|----------------------|--------------|
| <b>Random Forest</b> | <b>58.75</b> |
| Logistic Regression  | 58.02        |
| KNN                  | 57.66        |
| Naïve Bayes          | 53.28        |
| Decision Tree        | 54.01        |
| SVM                  | 56.00        |
| RBF Network          | 56.20        |
| AdaBoost             | 55.47        |
| <b>Bagging</b>       | <b>59.00</b> |

**Table 3.2.8 Accuracy values of Machine Learning Methods that have been performed on MIAS dataset**

Different feature selection methods have been performed and evaluated using the Random Forest classifier 10-fold CV, table 3.2.8 shows the results. Among all feature selection methods PCA performed best accuracy measure using random forest classifier.

| Feature Selection / Performance Metrics | Accuracy (%) | F-measure (%) |
|---|--------------|---------------|
| CFS                                     | 58.39        | 58.0          |
| Correlation Attribute Evaluation        | 56.20        | 55.2          |
| Gain Ratio                              | 58.02        | 57.3          |
| <b>Info Gain</b>                        | <b>60.94</b> | <b>60.2</b>   |
| <b>PCA</b>                              | <b>61.31</b> | <b>60.1</b>   |
| Relief Ranking                          | 56.56        | 55.5          |
| Symmetrical Uncertainty                 | 58.75        | 58.2          |
| <b>One Rule Attribute</b>               | <b>60.58</b> | <b>60.6</b>   |

**Table 3.2.9 Accuracy metrics of feature selection methods that have been performed on MIAS dataset using Random Forest classifier**

RBF Network, random forest and bagging classifiers have been evaluated using LOO-CV. The features are selected using PCA, parameters have been optimized. Table 3.2.9 describes the accuracy measures of this experiment. RBF Network outperforms to other classifiers.

| Feature Selection / Performance Metrics | Accuracy (%) | F-measure (%) |
|---|--------------|---------------|
| <b>RBF Network</b>                      | <b>77.41</b> | <b>75.86</b>  |
| Random Forest                           | 56.14        | 45.12         |
| Bagging                                 | 56.55        | 45.36         |

**Table 3.2.10 Classifiers have been evaluate using LOO-CV**

Instead of feature extracted dataset of MIAS, raw pixel data have been used as input of MLP. In order to do that, a new MIAS ROI database have been prepared, ROIs

were cropped in the same size which is equal to average width and height of the ROIs. Average accuracy of **56%±24%** has been obtained for 10-fold on **1 hidden layer MLP**. Moreover, the pixel dataset was used as the input of the simple **Convolution Neural Network** (CNN) and the accuracy of **68.9%** was obtained.

Image enhancement techniques increased the overall accuracy. The MIAS dataset is larger than the Pilot dataset, for this reason MLP can be implemented to this dataset. Simple MLP worked effectively and give the accuracy of 75%. Some optimization techniques have been applied in order to increase the accuracy, just ADAM optimization increased the accuracy gave 81.4%. For other cases the accuracy has been decreased or stayed the same. More than 1 hidden layer MLP was employed and the accuracy decreased 1%. Basic machine learning tools have been performed on MIAS dataset best accuracy of 59% and 58.75% is achieved using Bagging and Random forest algorithms. LOO-CV using RBF Network gave the highest result of 77% accuracy. Raw images have been used as input of MLP and CNN. 68.9% accuracy have been obtained by CNN.

# Chapter 4

## Conclusions and Future Prospects

### 4.1 Conclusions

In this thesis, a new feature extraction method is proposed for ROI detection in mammogram images. The proposed method first computes the Wavelet transform of the selected image, followed by extracting HOG and Haralick descriptors which compute textural and gradient features. Then the best set of features is selected using a wrapper strategy. When employed in a random forest classifier the proposed feature extraction method achieves the best ROI detection accuracy on the pilot dataset of the Digital Mammography DREAM Challenge whose image quality is high. Moreover, the hybrid feature extraction method developed in this thesis gives 8% better accuracy than the Haralick features. Different feature extractors have been used and results show that the classifier works better with combination of Wavelet, Haralick and HOG features. The parameter optimization and feature selection have increased the accuracy of ROI detection.

The proposed methods have also been applied to the MIAS dataset, which contains 322 images. Due to poor image quality and ambiguous tumorous regions, image enhancement techniques have been employed. The image enhancement methods increased the accuracy by 6.8%. Among the classification methods implemented, MLP worked effectively and gave an accuracy of 81.4% when the model is trained using the Adam optimization algorithm optimization. For other cases the accuracy has been decreased or stayed the same. Deep MLP models with more than one hidden layers have also been implemented but this decreased the accuracy by 1%. As an alternative to MLP models, raw images have been used as input to MLP and CNN however this did not perform better than the models that employ feature extraction methods. However if the CNN and other deep models are trained on larger datasets the prediction accuracy can be expected to be higher as demonstrated in Digital Mammography DREAM Challenge.

## 4.2 Future Prospects

As a future work, the proposed method can be used for automatic detection of ROIs in a CAD system. For this purpose, first, a mammogram image can be subdivided into small square-sized images by applying a sliding window. Then each image can be classified as ROI (positive) or not (negative). Second, the performance of the proposed method can be analyzed on other clinical databases (i.e DDSM) in order to verify the feasibility and adaptability of the results. Third, other feature extraction methods such as Curvelet can be implemented and tested on small and large datasets. Finally, deep learning methods and ensemble methods developed recently can be implemented and trained on larger image databases and combined with the existing method for breast cancer detection.



# BIBLIOGRAPHY

- [1] “What Is Cancer?” [Online]. Available: <https://www.cancer.org/cancer/cancer-basics/what-is-cancer.html>. [Accessed: 19-Oct-2018].
- [2] “What Are Tumors?” [Online]. Available: <http://pathology.jhu.edu/pc/BasicTypes1.php>. [Accessed: 19-Oct-2018].
- [3] F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre, and A. Jemal, “Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries,” *CA. Cancer J. Clin.*, Sep. 2018.
- [4] N. Szekely, N. Toth, and B. Pataki, “A Hybrid System for Detecting Masses in Mammographic Images,” *IEEE Trans. Instrum. Meas.*, vol. 55, no. 3, pp. 944–952, Jun. 2006.
- [5] M. Meselhy Eltoukhy, I. Faye, and B. Belhaouari Samir, “A statistical based feature extraction method for breast cancer diagnosis in digital mammogram using multiresolution representation,” *Comput. Biol. Med.*, vol. 42, no. 1, pp. 123–128, Jan. 2012.
- [6] J. Jinshan Tang, R. M. Rangayyan, J. Jun Xu, I. El Naqa, and Y. Yongyi Yang, “Computer-Aided Detection and Diagnosis of Breast Cancer With Mammography: Recent Advances,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 2, pp. 236–251, Mar. 2009.
- [7] “Breast Anatomy - National Breast Cancer Foundation.” [Online]. Available: <https://www.nationalbreastcancer.org/breast-anatomy>. [Accessed: 20-Oct-2018].
- [8] R. Highnam and M. Brady, *Mammographic Image Analysis*. Springer Netherlands, 1999.
- [9] H. Berment, V. Becette, M. Mohallem, F. Ferreira, and P. Chérel, “Masses in mammography: What are the underlying anatomopathological lesions?,” *Diagn. Interv. Imaging*, vol. 95, no. 2, pp. 124–133, Feb. 2014.
- [10] M. P. Morgan, M. M. Cooke, and G. M. McCarthy, “Microcalcifications Associated with Breast Cancer: An Epiphenomenon or Biologically Significant Feature of Selected Tumors?,” *J. Mammary Gland Biol. Neoplasia*, vol. 10, no. 2, pp. 181–187, Apr. 2005.
- [11] D. R. Ericeira, A. C. Silva, A. C. de Paiva, and M. Gattass, “Detection of masses

- based on asymmetric regions of digital bilateral mammograms using spatial description with variogram and cross-variogram functions,” *Comput. Biol. Med.*, vol. 43, no. 8, pp. 987–999, Sep. 2013.
- [12] Y. A. Reyad, M. A. Berbar, and M. Hussain, “Comparison of Statistical, LBP, and Multi-Resolution Analysis Features for Breast Mass Classification,” *J. Med. Syst.*, vol. 38, no. 9, p. 100, Sep. 2014.
- [13] A. Tahmasbi, F. Saki, and S. B. Shokouhi, “Classification of benign and malignant masses based on Zernike moments,” *Comput. Biol. Med.*, vol. 41, pp. 726–735, 2011.
- [14] G. B. Junior, E. C. da Silva, A. C. de Paiva, and A. C. Silva, “Breast Tissues Classification Based on the Application of Geostatistical Features and Wavelet Transform,” in *2007 6th International Special Topic Conference on Information Technology Applications in Biomedicine*, 2007, pp. 227–230.
- [15] S. Ergin and O. Kilinc, “A new feature extraction framework based on wavelets for breast cancer diagnosis,” *Comput. Biol. Med.*, vol. 51, pp. 171–182, Aug. 2014.
- [16] S. Beura, B. Majhi, and R. Dash, “Mammogram classification using two dimensional discrete wavelet transform and gray-level co-occurrence matrix for detection of breast cancer,” *Neurocomputing*, vol. 154, pp. 1–14, Apr. 2015.
- [17] R. Rouhi, M. Jafari, S. Kasaei, and P. Keshavarzian, “Benign and malignant breast tumors classification based on region growing and CNN segmentation,” *Expert Syst. Appl.*, vol. 42, pp. 990–1002, 2015.
- [18] N. Gedik, “A new feature extraction method based on multi-resolution representations of mammograms,” *Appl. Soft Comput.*, vol. 44, pp. 128–133, 2016.
- [19] S. Khan, M. Hussain, H. Aboalsamh, H. Mathkour, G. Bebis, and M. Zakariah, “Optimized Gabor features for mass classification in mammography,” *Appl. Soft Comput.*, vol. 44, pp. 267–280, Jul. 2016.
- [20] A. A. Shastri, D. Tamrakar, and K. Ahuja, “Density-wise two stage mammogram classification using texture exploiting descriptors,” *Expert Syst. Appl.*, vol. 99, pp. 71–82, Jun. 2018.
- [21] F. Pak, H. R. Kanan, and A. Alikhassi, “Breast cancer detection and classification in digital mammography based on Non-Subsampled Contourlet Transform (NSCT) and Super Resolution,” *Comput. Methods Programs Biomed.*, vol. 122,

- no. 2, pp. 89–107, Nov. 2015.
- [22] J. S. L. Jasmine, A. Govardhan, and S. Baskaran, “Microcalcification detection in digital mammograms based on wavelet analysis and neural networks,” *undefined*, 2009.
- [23] A. Mutaz, M. Abdalla, S. Dress, and N. Zaki, “Detection of Masses in Digital Mammogram Using Second Order Statistics and Artificial Neural Network,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 3, no. 3, 2011.
- [24] J. B. Jona and N. Nagaveni, “A Hybrid Swarm Optimization approach for Feature set reduction in Digital Mammograms.”
- [25] P. Gorgel, A. Sertbaş, N. Kilic, O. N. Ucan, and O. Osman, “MAMMOGRAPHIC MASS CLASSIFICATION USING WAVELET BASED SUPPORT VECTOR MACHINE.”
- [26] M. A. Berbar, “Hybrid methods for feature extraction for breast masses classification,” *Egypt. Informatics J.*, vol. 19, no. 1, pp. 63–73, Mar. 2018.
- [27] K. Bovis and S. Singh, “Detection of masses in mammograms using texture features,” *Proceedings-International Conf. Pattern Recognit.*, vol. 15, no. 2, pp. 267–269, 2000.
- [28] E. D. Pisano *et al.*, “Contrast Limited Adaptive Histogram Equalization image processing to improve the detection of simulated spiculations in dense mammograms,” *J. Digit. Imaging*, vol. 11, no. 4, pp. 193–200, Nov. 1998.
- [29] S. Jenifer, S. Parasuraman, and A. Kadirvelu, “Contrast enhancement and brightness preserving of digital mammograms using fuzzy clipped contrast-limited adaptive histogram equalization algorithm,” *Appl. Soft Comput.*, vol. 42, pp. 167–177, May 2016.
- [30] P. S. Heckbert, *Graphics gems IV*. AP Professional, 1994.
- [31] R. C. Gonzalez and R. E. (Richard E. Woods, *Digital image processing*. Prentice Hall, 2008.
- [32] J. P. Serra, *Image analysis and mathematical morphology*. Academic Press, 1982.
- [33] C. K. Chui, *An introduction to wavelets*. San Diego: Academic Press, 1992.
- [34] D. Wei *et al.*, “Classification of mass and normal breast tissue on digital mammograms: Multiresolution texture analysis,” *Med. Phys.*, vol. 22, no. 9, pp. 1501–1513, Sep. 1995.
- [35] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,”

- in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, 2005, vol. I, pp. 886–893.
- [36] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Textural Features for Image Classification,” *IEEE Trans. Syst. Man. Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.
- [37] “Specify Offset Used in GLCM Calculation - MATLAB & Simulink.” [Online]. Available: <https://www.mathworks.com/help/images/specify-offset-used-in-g lcm-calculation.html>. [Accessed: 08-Nov-2018].
- [38] “Gray level co-occurrence matrix · ImageFeatures.” [Online]. Available: <http://juliaimages.github.io/ImageFeatures.jl/latest/tutorials/g lcm.html>. [Accessed: 08-Nov-2018].
- [39] N. A. Hamilton, R. S. Pantelic, K. Hanson, and R. D. Teasdale, “(TAS) Fast automated cell phenotype image classification.” *BMC Bioinformatics*, vol. 8, p. 110, Mar. 2007.
- [40] “Local binary patterns - Wikipedia.” [Online]. Available: [https://en.wikipedia.org/wiki/Local\\_binary\\_patterns](https://en.wikipedia.org/wiki/Local_binary_patterns). [Accessed: 08-Nov-2018].
- [41] “Local Binary Patterns with Python & OpenCV - PyImageSearch.” [Online]. Available: <https://www.pyimage search.com/2015/12/07/local-binary-patterns-with-python-opencv/>. [Accessed: 08-Nov-2018].
- [42] P. B. Rao, D. V. Prasad, and C. Pavan Kumar, “Feature Extraction Using Zernike Moments.”
- [43] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.
- [44] M. A. Hall, “Correlation-based Feature Selection for Machine Learning,” 1999.
- [45] “CorrelationAttributeEval.” [Online]. Available: <http://weka.sourceforge.net/doc.dev/weka/attributeSelection/CorrelationAttributeEval.html>. [Accessed: 09-Nov-2018].
- [46] J. Novakovic, “Using Information Gain Attribute Evaluation to Classify Sonar Targets.”
- [47] R. J. Urbanowicz, M. Meeker, W. LaCava, R. S. Olson, and J. H. Moore, “Relief-Based Feature Selection: Introduction and Review,” Nov. 2017.
- [48] B. Kushwaha and N. Vyas, “A Feature Subset Selection Technique for High Dimensional Data Using Symmetric Uncertainty,” *J. Data Anal. Inf. Process.*, vol. 2, pp. 95–105, 2014.
- [49] N. El Aboudi and L. Benhlima, “Review on wrapper feature selection

- approaches,” in *2016 International Conference on Engineering & MIS (ICEMIS)*, 2016, pp. 1–5.
- [50] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*. Pearson Addison Wesley, 2005.
- [51] J. R. (John R. Quinlan and J. Ross, *C4.5 : programs for machine learning*. Morgan Kaufmann Publishers, 1993.
- [52] J. R. Quinlan, “Induction of Decision Trees,” 1986.
- [53] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*. Springer, 2013.
- [54] Tin Kam Ho, “Random decision forests,” in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, pp. 278–282.
- [55] Tin Kam Ho, “The random subspace method for constructing decision forests,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, 1998.
- [56] A. Gelzinis *et al.*, “Exploring sustained phonation recorded with acoustic and contact microphones to screen for laryngeal disorders,” *IEEE SSCI 2014 - 2014 IEEE Symp. Ser. Comput. Intell. - CICARE 2014 2014 IEEE Symp. Comput. Intell. Healthc. e-Health, Proc.*, no. December, pp. 125–132, 2015.
- [57] N. S. Altman, “An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression,” *Am. Stat.*, vol. 46, no. 3, pp. 175–185, Aug. 1992.
- [58] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.
- [59] D. Coomans and D. L. Massart, “Alternative k-nearest neighbour rules in supervised pattern recognition,” *Anal. Chim. Acta*, vol. 136, pp. 15–27, 1982.
- [60] “K-nearest neighbors.” [Online]. Available: [https://en.wikipedia.org/wiki/K-nearest\\_neighbors\\_algorithm](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm). [Accessed: 07-Oct-2018].
- [61] S. J. (Stuart J. Russell, P. Norvig, and J. Canny, *Artificial intelligence : a modern approach*, 2nd ed. Prentice Hall, 2003.
- [62] “Naive Bayes for Machine Learning.” [Online]. Available: <https://machinelearningmastery.com/naive-bayes-for-machine-learning/>. [Accessed: 08-Oct-2018].
- [63] “Naive Bayes Classifier.” [Online]. Available: [https://en.wikipedia.org/wiki/Naive\\_Bayes\\_classifier](https://en.wikipedia.org/wiki/Naive_Bayes_classifier). [Accessed: 08-Oct-2018].
- [64] S. H. Walker and D. B. Duncan, “Estimation of the Probability of an Event as a Function of Several Independent Variables,” *Biometrika*, vol. 54, no. 1/2, p. 167,

- Jun. 1967.
- [65] D. W. Hosmer and S. Lemeshow, *Applied logistic regression*, 2nd ed. Wiley, 2000.
- [66] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the fifth annual workshop on Computational learning theory - COLT '92*, 1992, pp. 144–152.
- [67] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.
- [68] G. Hughes, “On the mean accuracy of statistical pattern recognizers,” *IEEE Trans. Inf. Theory*, vol. 14, no. 1, pp. 55–63, Jan. 1968.
- [69] B. Kégl, “The return of AdaBoost.MH: multi-class Hamming trees,” *arXiv*, Dec. 2013.
- [70] F. ROSENBLATT, “The perceptron: a probabilistic model for information storage and organization in the brain,” *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, Nov. 1958.
- [71] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.
- [72] “ANN.” [Online]. Available: [https://en.wikipedia.org/wiki/Artificial\\_neural\\_network](https://en.wikipedia.org/wiki/Artificial_neural_network). [Accessed: 09-Oct-2018].
- [73] S. S. Haykin, *Neural networks : a comprehensive foundation*, 2nd ed. Prentice Hall, 1999.
- [74] T. Isokawa, H. Nishimura, N. Matsui, T. Isokawa, H. Nishimura, and N. Matsui, “Quaternionic Multilayer Perceptron with Local Analyticity,” *Information*, vol. 3, no. 4, pp. 756–770, Nov. 2012.
- [75] F. Schwenker, H. A. Kestler, and G. Palm, “Three learning phases for radial-basis-function networks,” *Neural Networks*, vol. 14, no. 4–5, pp. 439–458, May 2001.
- [76] M. D. (Martin D. Buhmann, *Radial basis functions : theory and implementations*. Cambridge University Press, 2003.
- [77] “Radial Basis Function Network (RBFN) Tutorial · Chris McCormick.” [Online]. Available: <http://mccormickml.com/2013/08/15/radial-basis-function-network-rbfn-tutorial/>. [Accessed: 10-Oct-2018].
- [78] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning

- applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [79] “An intuitive guide to Convolutional Neural Networks.” [Online]. Available: <https://medium.freecodecamp.org/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050>. [Accessed: 10-Oct-2018].
- [80] “A Gentle Introduction to k-fold Cross-Validation.” [Online]. Available: <https://machinelearningmastery.com/k-fold-cross-validation/>. [Accessed: 10-Oct-2018].
- [81] R. O. Duda, P. E. (Peter E. Hart, and D. G. Stork, *Pattern classification*. Wiley, 2001.
- [82] S. V. Stehman, “Selecting and interpreting measures of thematic classification accuracy,” *Remote Sens. Environ.*, vol. 62, no. 1, pp. 77–89, Oct. 1997.
- [83] D. M. W. Powers, “Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation,” *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.
- [84] “Confusion Matrix.” [Online]. Available: [https://en.wikipedia.org/wiki/Confusion\\_matrix](https://en.wikipedia.org/wiki/Confusion_matrix). [Accessed: 10-Oct-2018].
- [85] M. Hossin and Sulaiman, “A REVIEW ON EVALUATION METRICS FOR DATA CLASSIFICATION EVALUATIONS,” *Int. J. Data Min. Knowl. Manag. Process*, vol. 5, no. 2, 2015.
- [86] “classification - What does AUC stand for and what is it? - Cross Validated.” [Online]. Available: <https://stats.stackexchange.com/questions/132777/what-does-auc-stand-for-and-what-is-it>. [Accessed: 11-Oct-2018].
- [87] S. B. Yengec Tasdemir, K. Tasdemir, and Z. Aydin, “ROI Detection in Mammogram Images using Wavelet-Based Haralick and HOG Features.”
- [88] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA Data Mining Software: An Update.”
- [89] M. Abadi *et al.*, “TensorFlow: A system for large-scale machine learning.” 2016.