Breast Cancer Detection in Mammogram Medical Images with Data Mining Techniques

Konstantinos Kontos, Manolis Maragoudakis

Breast Cancer Detection in Mammogram Medical Images with Data Mining Techniques

Konstantinos Kontos, Manolis Maragoudakis

University of the Aegean,
Department of Information and Communication Systems Engineering,
Samos, Greece

{icsdm11027,mmarag}@aegean.gr

Abstract. A domain of interest for data mining applications is the study of biomedical data which, in combination with the field of image processing, provide thorough analysis in order to discover hidden patterns or behavior. Towards this direction, the present paper deals with the detection of breast cancer within digital mammography images. Identification of breast cancer poses several challenges to traditional data mining applications, particularly due to the high dimensionality and class imbalance of training data. In the current approach, genetic algorithms are utilized in an attempt to reduce the feature set to the informative ones and class imbalance issues were also dealt by incorporating a hybrid boosting and genetic sub-sampling approach. As regards to the feature extraction approach, the idea of trainable segmentation is borrowed, using Decision Trees as the base learner. Results show that the best precision and recall rates are achieved by using a combination of Adaboost and k-Nearest Neighbor.

Keywords: Image Processing, Trainable Segmentation, Data Mining, Genetic Algorithms.

1 Introduction

The term breast cancer refers to the development of malignant tumor in the breast. It is one of the most common cancers worldwide and accounts for 22.9% of all cancers (excluding non-melanoma skin cancers) in women. Caused by uncontrolled proliferation of pathological cells, breast cancer results in the formation of malignant tumor [1]. Pathological cells have the potential of spreading to adjacent tissues in hostile consequences for the entire organization. Conversely, the incidence of the disease in males is real but very small. Unfortunately, little are known about the causes of breast cancer, despite the fact that researchers have identified several risk factors, such as age, heredity, disorders of menstruation, alcohol consumption, obesity, smoking, contraceptive pills, history of cancer, exposure to radiation and sedentary life. A very important method of detecting breast cancer is mammography. The goal of mammography is the early detection of breast cancer, typically through detection of characteristic masses and/or microcalcifications. It can be done either by the classical method of radiography or digital mammography. In any case, the doctor's opinion is necessary, based on the findings, so if the results of mammography indicate a tumor further tests must be done.

Based on the outcome of this method, this paper proposes a classification system for identifying segments of cancer cells from mammography images in order to assist professional radiologists and gynecologists in their diagnosis. Emphasis has been given to the feature extraction process, in which we borrow the idea of [2] and apply a trainable image segmentation process. Since breast cancer cases are much rarer than healthy ones, it is evident that the classification problem will face a large imbalance between the two classes. In the discussed domain, the percentage of the positive class (i.e. tumor) is only 2.6%, therefore, traditional classification approaches are very likely to fail towards producing a robust model, able to generalize well for future, unseen cases.

Apart from the class imbalance problem, there is also a large number of input features, obtained from the feature extraction phase. Since not all of them are actually informative for the class separation process, a feature selection approach is followed, using Genetic Algorithms and a performance metric as fitness function. Boosting is also used together with a deterministic filter in order to eliminate noisy and redundant instances from the majority class. Experimental results support our claim that feature selection and sub-sampling using Genetic Algorithms and boosting are able to result in a robust and accurate breast cancer identification system, providing classification outcomes in terms of precision and recall that are within the current State-of-the-Art.

The structure of the article is as follows: in Section 2, previous work is described in order to obtain a general perspective of the task at hand. In Section 3, the pre-processing phase is analyzed, with emphasis given to the segmentation and feature extraction steps. In Section 4, a theoretical background of Genetic Algorithms, Adaboost and k-NN is provided. In Section 5, the detailed description of the proposed approach is contained, along with the experimental results. Finally, in section 6, the conclusions are presented, followed by the acknowledgements.

2 Previous Work

Throughout recent years, there have been several studies the field of data mining in oncology:

- The work of [3] presented a model that was designed to automatically detect breast cancer through a Bayesian network. In this article, the number of dataset instances (taken from Wisconsin breast cancer database)[1] was decreased, in order to improve results. For the reduction of the dataset, the technique of feature ranking was applied, thus, features that were not important were eliminated. The correct classification rate that achieved by using Bayesian networks was 98.15%. Although the performance is quite good, the system has some evident weaknesses. At first, the large amount of preprocessing steps along with the inherent slow learning rate of Bayesian networks result in a fairly slow process. Moreover, the dataset is not automatically extracted from mammogram photos but used the Wisconsin breast cancer database. This fact can cause variations in system performance, if the attributes of mammogram photos that has to be tested, are quite different from the Wisconsin dataset. In other words, this technique cannot be used in photos with different attributes from the database of Wisconsin.

- The article presented by Rani [4] dealt with the creation of a system that automatically detects breast cancer with neural networks, executed in a parallel manner. The experimental results showed that the best accuracy of the model is 92% in a multilayer neural network, with 300 training samples and 50 test samples. As we shall see in later paragraphs, this outcome is far lower than that of our proposed work.

- The paper of Pin Wei and Ming Der [5], presented three different algorithms for detecting breast cancer and essentially highlights the fact that the genetic algorithm can provide better results than other techniques. The accuracy of the proposed algorithms was achieved by the method of 10-fold cross validation. The first algorithm is a variation of decision trees. The average correct classification rate is 94.35%. Regarding the neural network algorithm, its performance is slightly higher than that of decision trees, in the range of 95.02%. In contrast to these two algorithms, the genetic algorithm that applied in the present paper provides not only better results but also a thorough analysis of class-imbalance and feature selection process. Additionally, another drawback of this approach is that dataset is not automatically extracted from mammogram photos but used the Wisconsin breast cancer database, as in the paper of [3]. As denoted above, this fact can cause variations in system performance, if the attributes of mammogram photos that has to be tested, are quite different from the Wisconsin dataset.

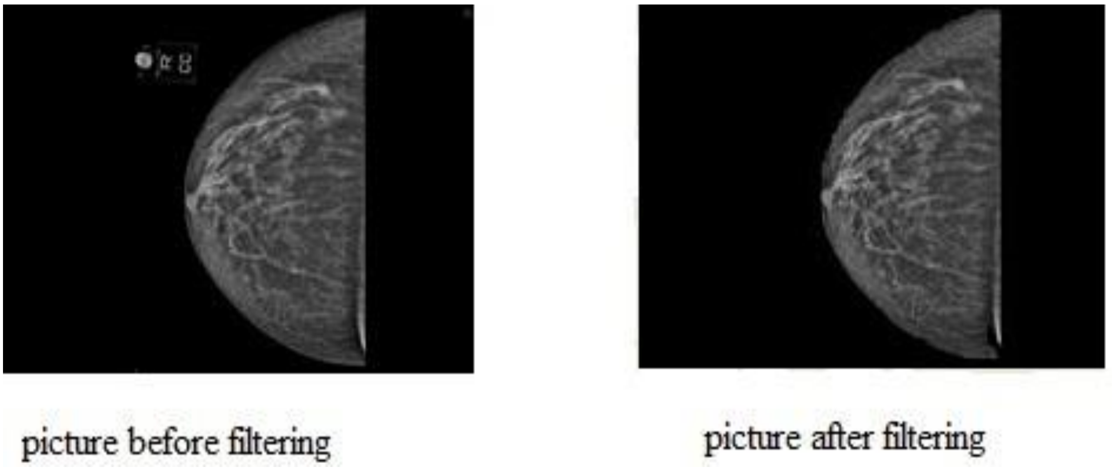
A general deduction from the above studies is that performance should not be measured as the total number of correct predictions. While in many classification tasks this may seem reasonable, for the task at hand, where examples of the healthy class overwhelm the examples of the cancer class, this

[1] http://orange.biolab.si/doc/datasets/breast-cancer-wisconsin-cont.htm

performance metric is not informative since one can obtain good performance simply by favoring the majority class.

3 Data Pre-processing

In order to perform identification of breast cancer, a set of 45 mammography images was kindly provided by a private diagnostic center, upon having manually annotated malignant tumors by a specialist. Images were cleaned in order to remove handwritten codes that were found within (e.g. type of the chest, image code, etc.). Since these types of annotations varied in size and position, an automated filtering approach was adopted, which operated as follows: initially, a Gaussian blur filter was applied to ensure that spurious high-frequency information does not appear again and smooth the image. The Statistical Region Merging (SMR) algorithm [6] was then used to extract segments within the image. After several experiment with various bin sizes, a small set of the resulted segments was labeled as those that contained the handwritten annotations and a decision tree learner (using Information Gain as splitting criterion) was used in order to classify the rest of the segments for all of the remaining training images. Segments that were labeled as containing the handwritten annotations were eliminated (masked as black to be more precise) and the remaining segments were merged together so that a clean image was produced. Fig. 1 depicts a sample of an original and a cleaned image. It is noteworthy to mention that the SMR algorithm together with the Decision Tree classifier achieved 100% accuracy, meaning that in all of the input images, the area with the annotations was successfully identified and eliminated without any human interference. In the next sub-sections, a detailed analysis of the image segmentation and feature extraction methodologies is provided.



picture before filtering

picture after filtering

Fig. 1. An example of the cleaning process, using statistical region merging for segmentation and decision trees for identifying segments that contained manual annotations.

3.1 Image Segmentation

As described in the introductory section, upon cleaning of the images, a segmentation step was carried out, by using trainable segmentation with low-level features, an approach initially inspired by [2].

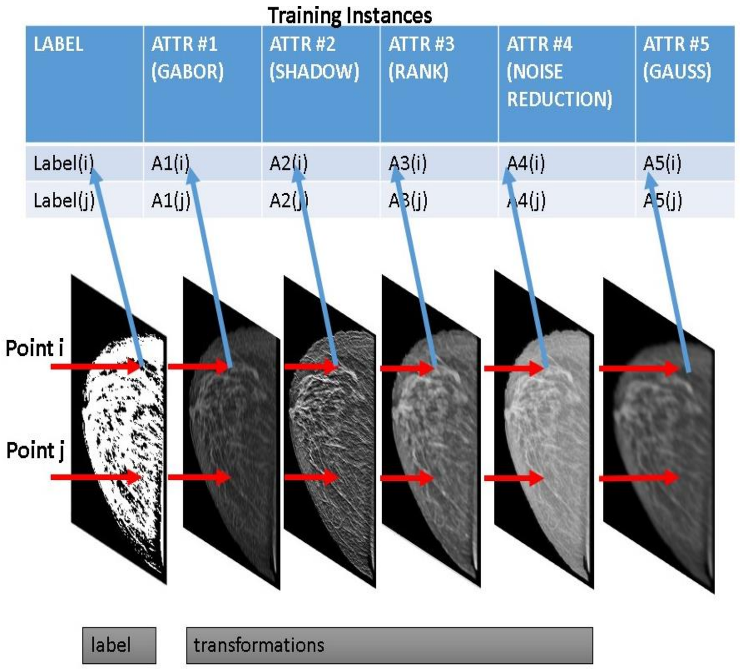


Fig. 2. The main context of trainable segmentation.

The exact procedure contains the following steps:

- Histograms of all images were equalized in order to diminish variation in intensity between layers.
- Points from both classes (*healthy* and *tumor*) were selected, giving importance to doubtful areas where healthy tissue is resembling problematic cases and vice-versa (Fig. 3).

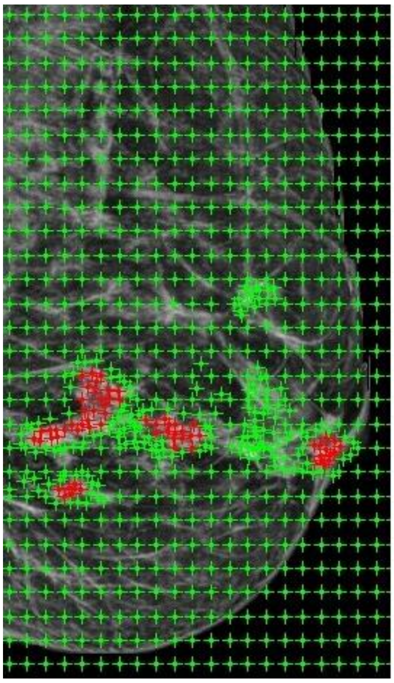


Fig. 3. Selection of training points. Green points denote healthy and red points denote tumor tissues.

- Extract features from each selected by applying several transformations in a multi-layered fashion (see Fig. 2). The majority of the aforementioned filters use the value of each local pixel and also the values of surrounding pixels in order to be computed. Thus, by using a variety of transformations, it is possible to extract a more complex perspective about each image area. In our case, we applied Gaussian blur filters with σ varying from 2 to 16, image convolution transform using the north shadow kernel, a Gabor filter using a radius of 4 and wavelength equal to 8, a Rank filter with type set to mean and radius to 4 and finally a noise reduction filter. All of the considered transforms were obtained from the ImageMining extension[2] of RapidMiner®[3] and their exact description is beyond the scope of this article. For a detailed explanation of them, please refer to [2].

- Train the segmentation algorithm using Decision Trees with Gain Ratio as splitting criterion.

3.2 Feature Extraction

The completion of the former phase resulted in creating segments from each training image. Segments were manually evaluated using a hand-labeled set of ground-truth segmentations. The measure of comparison was the Rand index, which is commonly used in statistics Segment-level features were considered based on color and shape. The following table tabulates the parameters obtained from each Region of Interest (ROI). Furthermore, an additional set of 64 features was also extracted by calculating the histogram of each segment. Therefore, the total number of extracted features was 64+18=82. The final datasets, upon segmentation and feature extraction contained about 35.000 examples, 920 of them (an analogy of 2.6%) belonging to the tumor class and the rest belonging to the healthy class, with 82 attributes. It is clear that the presented dataset poses significant challenges to traditional classification algorithms, due to the class imbalance issues and also due to the large (however not huge) number of input attributes. The proposed methodology uses Genetic algorithms to deal with feature selection.

Table 1. Segment-Based Features

Name	Comments
Mean	The average gray value in the image.
Kurtosis	The fourth order moment about the mean.
Skewness	The third order moment about the mean.
Standard deviation	Standard deviation of the gray values used to generate the mean gray value.
Min/max gray value	
Area	Area of ROI in pixels.
Area fraction	The percentage of non-zero pixels.
Center of mass	the brightness-weighted average of the x and y coordinates of all pixels in the image.
Centroid of ROI	
Major/Minor Axis	Length of major/minor axis of fitted ellipse.
Angle	Angle in degrees of fitted ellipse.
Perimeter	The length of the outside boundary of the selection.
Feret /Projected Area diameter	
Circularity	4pi(area/perimeter^2).
Eccentricity	

[2] http://splab.cz/en/research/data-mining/articles
[3] http://www.rapid-i.com

4 Theoretical Background.

4.1 Genetic Algorithms

The basic idea behind Genetic Algorithms (GA) is the imitation of the mechanisms of nature. The basic idea behind Genetic Algorithms (GA) is the imitation of the mechanisms of nature. Take, for example, hares and how they reproduce and evolve from generation to generation. Suppose we begin to observe a certain population of hares. Naturally, some of them will be faster and glibber than others. These faster and smarter hares are less likely to be the meal of a fox and so by the time they manage to survive, will deal with reproduction of their kind. Of course, there will be a small number of slow and less glib hares, who will manage to survive just because they were lucky. All these hares that have managed to survive, will begin production of the next generation, a generation that combines all the features of its members, combined in various ways to each other. So, slow hares will mixed with some fast, some fast with other fast, and some glib hares with less glib and so on. The hare of the next generation will be, on average, faster and smarter than their ancestors since from the previous generation survived more quick and clever hares. Fortunately, for the preservation of the natural balance, foxes replaced in the same replication process, otherwise the hares will become too fast and smart to be caught. By analogy with living creatures, referred to persons or genotype in a population. Very often these persons also called chromosomes. This can lead to some wrong conclusions if a parallel with the natural organisms, where each cell of any specific type containing a specified number of chromosomes (human cells, for example containing 46 chromosomes). In GA almost always refer to individuals with a single chromosome. The chromosomes that are composed of different elements called genes and are arranged in linear sequence. Each gene affects the inheritance of one or more characteristics. Genes affecting specific features of the individual person and in specific positions of a chromosome called places (loci). Each feature of the individual (such as the hair color) has the ability to display in various forms, depending on the situation in which the corresponding gene that affects. These different situations that can get the gene, called alleles (attribute values). Each genotype (which in most cases is only one chromosome) represents a possible solution to a problem. The translated content of a given chromosome is called phenotype and specified by the user, depending on their needs and requirements. A development process applied on a population of chromosomes represents an extensive search in a space of possible solutions. A prerequisite for the successful outcome of such balancing is groping two processes are obviously contradictory, exploitation and conservation of the best solutions and the best possible exploration of the whole space.

A GA searches in many directions by maintaining a population of potential solutions and by supporting recording and exchanging information between these orientations. The population undergoes a simulated genetic evolution. In each generation, relatively "good" solutions reproduce, while relatively "bad" removed. The separation and evaluation of different solutions achieved by means of a fitness function), which plays the role of the environment in which the population evolves [7]. A GA for a given problem should be composed of the following steps:

1. Generate an initial population consisting of population_size individuals. Each attribute is switched on with probability p_initialize.
2. For all individuals in the population
   - Perform mutation, i.e. set used attributes to unused with probability p_mutation and vice versa.
   - Choose two individuals from the population and perform crossover with probability p_crossover. The type of crossover can be selected by crossover_type.
3. Perform selection, map all individuals to sections on a roulette wheel whose size is proportional to the individual's fitness and draw population size individuals at random according to their probability.
4. As long as the fitness improves, go to step 2.

4.2 Adaboost

Boosting [8], is a Machine Learning iterative process that can adaptively change the distribution of selected training examples, focusing on those that are particularly difficult to be classified. As mentioned before, for the task at hand, examples of the positive class are much less than the negative and thus, traditional learners fail to classify them correctly. That is exactly where boosting contributes, i.e. to change the weights (increase) of instances that are difficult to be correctly labeled and then favor the selection of them on the next iteration using these weights. The most representative implementation of this process is the Adaboost [9] algorithm. Adaboost creates training sets by sampling with replacement, according to a weighting factor W. Initially, all instances are equally weighted. In each iteration, a base classifier (also called a weak learner) is used for training and testing, and the error rate is calculated for each testing instance. Those that are correctly classified lower their weights and those that are not increase their weight, so that in the next round, they are more probable to be selected. The final outcome of Adaboost is a majority voting over all weak learners trained before. The exact process is mentioned below:

```
1. Initialize all weight equally (w=1/N, where N is the number of instances)
2. Let T be the number of iterations.
3. For i=1 to T do:
   (a) Create a training set Di by sampling (with replacement) from all available examples, according to w.
   (b) Train a base classifier Ci on Di.
   (c) Apply Ci to all examples in the original dataset D.
   (d) Calculate the error, $\varepsilon_i = \frac{1}{N}\left[\sum_j w_j \delta(C_i(x_j) \neq y_j\right]$ (where $\delta(C_i(x_j) \neq y_j$ equals to 1 if the prediction is correct and 0 otherwise).
   (e) If εi>0.5 then
       (i) Reset all weights for all Examples.
       (ii) Go back to step 3a.
   (f) End if
   (g) $a_i = \frac{1}{2}\ln\frac{1-\varepsilon_i}{\varepsilon_i}$
   (h) Update weights according to equation: $w^{(j+1)=\frac{{w_i}^{(j)}}{z_j}\times\left\{\begin{matrix} exp^{-\alpha j}if\ classified\ correctly \\ exp^{aj}if\ classidied\ incorrectly \end{matrix}\right.}$
   (i) End for
```

- Output the prediction as: $C(x) = argmax(\sum_{j=1}^{T} a_j \delta(C_i(x) = y))$.

4.3 k-Nearest Neighbor

The k-Nearest Neighbor algorithm (k-NN) is a method for classifying objects based on the closest training examples in the feature space. k-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification [10]. The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of its nearest neighbor. Figure 4 depicts the basic operation of the algorithm.

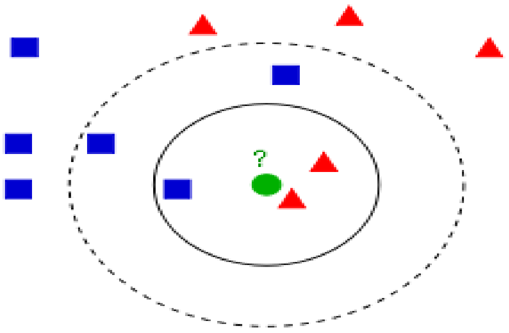


Fig. 4. An Example of k-NN classification. The test instance (green circle) should be labelled either as red triangle or blue squares. If K equals 3 (the solid circle) it is assigned to the first class, because there are 2 triangles and only 1 square inside the inner circle. If K equals 5 (the dashed line circle) it is assigned to the second class, since 3 squares outperform the 2 triangles inside this circle.

The neighbors are taken from a set of objects for which the correct classification is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. The k-nearest neighbor algorithm is sensitive to the local structure of the data. Usually Euclidean distance is used as the distance metric; however this is only applicable to continuous variables. In cases such as text classification, another metric such as the overlap metric (or Hamming distance [11]) can be used. Often, the classification accuracy of "k"-NN can be improved significantly if the distance metric is learned with specialized algorithms such as Large Margin Nearest Neighbor [12] or Neighborhood Component Analysis [13].

5 Methodology and Experimental Results

As mentioned above, the class imbalance problem was dealt using Adaboost and k-NN with k=3 as a weak learner. Since Adaboost is sensitive to noisy data, a filtering approach was applied though a 2D visual inspection of the training points that resulted in the elimination of more than 15% of the majority class instances. A set of 10 iterations for Adaboost proved to balance between both complexity and accuracy. In order to deal with the feature selection problem, a genetic approach was used to find the best subset of input features. The representation of genes was straightforward, in the sense that a bit of 0 in the j-th position symbolized the absence of the j-th input feature and vice-versa. The f-measure [14] metric was chosen as fitness function, since accuracy may always favor the majority class and is not a clear depicter of the classification significance. A population size of 15 was used and selection was performed by Boltzman tournament [15]. The maximum number of generations was set to 50. Probability of mutation was set to 1/100 and probability of crossover (uniform) was set to 0.5. Experiments were carried out using the 10-fold cross validation method. In the following tables, the performance was measured in terms of precision P and recall R (or Sensitivity) for both classes. Precision is given by: $P = \frac{tp}{tp+fp}$, while recall equals to: $R = \frac{tp}{tp+fn}$. The first experiment aims at denoting the usefulness of the boosting process, which confronts imbalance data effectively. Therefore, in the following table, results obtained from a set of basic classifiers, namely, Decision Trees, k-NN using k=3 and Naïve Bayes are tabulated.

Table 2. Precision and Recall metrics for the base classifiers.

	Decision Trees		**k-NN**		**Naïve Bayes**	
	Precision	Recall	Precision	Recall	Precision	Recall
Healthy Class	97.68%	98.05%	98.75%	98.4%	99.3%	91.67%
Tumor Class	71.79%	69.23%	74.87%	78.6%	45.61%	91.97%

As seen by the results, recall and precision for the tumor class are quite low, which is not adequate for a robust tumor identification/decision support system. By using the proposed boosting strategy, even without the inclusion of the feature selection module, results are far better. In order to depict the contribution of the feature selection genetic approach, results with and without this module are tabulated in Tables 3 and 4 respectively:

Table 3. Performance of the system using Adaboost and k-NN, without feature selection.

Total Accuracy: 97,32%	**Precision**	**Recall**
Healthy Class	98.73%	98.38%
Tumor Class	79.55%	83.28%

Table 4. Performance of the system using Adaboost and k-NN, with Genetic feature selection.

Total Accuracy: 99,24%	Precision	Recall
Healthy Class	99.69%	97.28%
Tumor Class	99.53%	88.29%

As seen from the above tables, performance is significantly improved in both precision and recall for the positive class (tumor). The genetic algorithm feature selection subsystem results in an increase of 20% in precision and 5% in recall for the minority class. The total accuracy reaches 99.24% which is, according to our knowledge, is within the current state of the art in the field and slightly better. Nevertheless, results need to be compared to a common set of training and testing mammograms in order to fully support the previous claim.

6 Conclusion and future work

One of the best methods of detecting breast cancer is mammography but in some cases, radiologists cannot clearly detect tumors despite their experience. Based on the utility of this method, this paper proposes a system for detecting cancer that could assist medical staff and improve the accuracy of identification of tumor tissues. In the current approach, two main problems were faced. The former deals with class imbalance within the training set, which in our case reached a tiny 2.6% of the rare class distribution compared to the majority class. The latter deals with a plethora of extracted input features and the need for a selection scheme over them. Towards the first direction, a boosting approach was utilized, along with manual removal of noisy examples with k-NN learner. For the second issue, a Genetic feature selection method was incorporated which led to a significant improvement in precision and recall metrics for the minority class, while maintaining high performance scores for the majority class as well. An accuracy of 99.24% was reached, in a dataset of 45 mammograms that generated approximately 11.500 instances. Results were obtained using a 10-fold cross validation methodology. In the future, we intend to use a larger mammographic database (with more than 100 mammograms) and to extract more features from the images. More specifically, features that can be extracted from the images such as dilate (increases the size of bright objects) etc., could be of particular interest and relevant for classification. The influence and the performance of these new attributes will be thoroughly studied. An additional future research direction is the study of parameter selection for the K-NN algorithm as well as the choice of the distance metric.

Acknowledgement

The authors would like to thank the Diagnostic Center, Magnitiki of Patras for the ongoing support and free access to digital mammography photos and the radiologist Mrs. Kokoli Maria for her valuable collaboration and her continuous assistance she provided throughout the annotation and evaluation phase.

References

1. Altman, M.B., Flynn, M.J., Nishikawa, R.M., Chetty, I.J., Barton, K.N., Movsas, B., Kim, J.H., Brown, S.L.: The potential of iodine for improving breast cancer diagnosis and treatment, Medical Hypotheses. vol. 80, Issue 1, (2013).
2. Burget, R., Uher, V., Masek, J.: Trainable Segmentation Based on Local-level and Segment-level Feature Extraction. ISBI conference (2012).

3. Fallahi, A., Jafari, S.: An Expert System for Detection of Breast Cancer Using Data Preprocessing and Bayesian Network. International Journal of Advanced Science and Technology vol. 34, (2011).
4. Rani, U.K.: Parallel Approach for Diagnosis of Breast Cancer using Neural Network Technique. International Journal of Computer Applications, vol. 10, no. 3, (2010).
5. Chang, W.P, Liou, D.L.: Comparison of Three Data Mining Techniques with Genetic Algorithm in the Analysis of Breast Cancer Data.
6. Nock, R., Nielsen, F.: Statistical Region Merging. IEEE Transactions of pattern analysis and machine intelligence, vol. 26, no. 11, (2004).
7. Lykothanasis, S.: Genetic Algorithms and Applications (Greek: Γενετικοί αλγόριθμοι και εφαρμογές), (2001).
8. Schapire, R.E., Freund, Y.: Boosting: foundations and algorithms, (2012).
9. Freund, Y., Schapire, R.E.: A Short Introduction to Boosting.Journal of Japanese Society for Artificial Intelligence. (In Japanese, translation by Naoki Abe.), (1999).
10. Bremner, D., Demaine, E., Erickson, J., Iacono, J., Langerman, S., Morin, P., Toussaint, G.: Output-sensitive algorithms for computing nearest-neighbor decision boundaries. Discrete and Computational Geometry, (2005).
11. Hamming, Richard, W.: Error detecting and error correcting codes. Bell System Technical Journal 29, (1950).
12. Weinberger, Blitzer, K.Q, Saul, L.K.: Distance Metric Learning for Large Margin Nearest Neighbor Classification. Advances in Neural Information Processing Systems, NIPS, vol.18, (2006).
13. Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R.,: Neighbourhood Component Analysis. Advances in Neural Information Processing Systems. vol. 17, pp. 513-520, (2005).
14. Hripcsak, G., Rothschild, A.S.: Agreement, the F-Measure, and Reliability in Information Retrieval. American Medical Informatics Association (2005).
15. Goldberg, D.E.: A Note on Boltzmann Tournament Selection for Genetic Algorithms and Population-Oriented Simulated Annealing, Complex Systems. vol. 4, pp. 445-460 (1990).