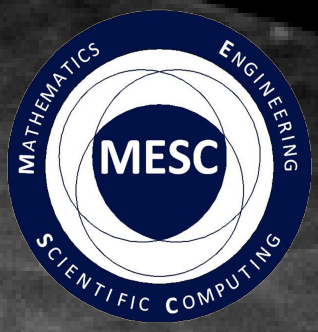


LIBCAD

Call_It_Anywhere();

Technical Documentation
Ver. 2.0 beta

SHARED LIBRARY AND CLOUD-BASED API
FOR ANY DIGITAL MAMMOGRAPHY
FOR ANY IMAGE VIEWER

A COMPLETE GUIDE TO LIBCAD
1- DLL AND SHARED OBJECT LIBRARY
2-CLOUD-BASED API

Acknowledgement: MESC for Research and Development and Waleed A. Yousef are grateful for Professor Naglaa Abdelrazek, of Cairo University and AlfaScan Radiology Center, for working as the consultant radiologist in an early stage of this project.

MESC for Research and Development
web: `www.MESCLABS.com`
email: `info@MESCLABS.com`

LICENSING

LIB License: is currently the only available licensing option for LIBCAD. This is almost suitable for companies producing image viewers and wishing to integrate the CAD capabilities of LIBCAD to their viewer. The installation CD is issued for a particular machine ID. It cannot be installed on another machine. If your machine ID changes, due to hardware upgrading, you just need to inform us to accomodate these changes without needing to reinstall. Please, email us to send you `MachineID.exe` to run it on the target machine on which LIBCAD will be running. `MachineID.exe` will display your machine ID; please, send it to us and we will be able to issue the installation CD for this particular machine.

Web License: is a future license for a stand alone user interface for an image viewer along with CAD capabilities; yet without PACS service. This will be a convenient alternative to researcher radiologists needing a CAD tool on their laptops without necessarily being connected to digital mammography.

PREFACE

Why this software? This is a normal question that rises specially when we know that Computer Aided Detection (CAD) software (specially for breast cancer detection) has been around for several years. Some CADs are even approved for commercial use by regulatory agencies, e.g., the U.S. Food and Drug Adminstration (FDA). However, those commercial CADs are too expensive so that they are not affordable for any laboratory in many countries.

Our objective for this CAD library can be summarized in two words: “affordable” and “widespread”. ”Affordable” means it is much less expensive than other commercial CADs. “Widespread”, means it is offered in the form of a Dynamic Link Library (DLL) so that any one can call it using any programming language. Many companies existing in the market design their commercial image viewers, yet without CAD capabilities. This is where our LIBCAD comes to the picture.

LIBCAD is designed to be a library to be called by any programmer whether developing an image viewer and GUI, doing a batch processing, or even using the library in scientific research and pure scientific computing. Therefore, the functions in this library produce an output suitable for any application, leaving the interfacing issues to the designer of the application of interest.

The current version, Ver. 2.0 beta, is our first release. By the moment of writing this manual and preparing for the release of this first version, our R&D team is developing a much better version in terms of detection accuracy. Your input and feedback are highly appreciable and valuable for us for continuous improvement.

We would like to take the opportunity and acknowledge the Information Technology Industry Development Agency (ITIDA) for its role in supporting scientific research. ITIDA is an Egyptian organization for promoting industry-academia collaboration. LIBCAD started in 2010 as a research project funded by ITIDA. The academic partner of this project was Waleed A. Yousef, Ph.D. of Helwan University. The industrial partner was MESC for Research and Development.

CONTENTS

CHAPTER 1

INTRODUCTION

This is a technical documentation for using the LIBCAD library. For the Shared Library, we assume that the user is a technical developer who is experienced in programming under Windows in any programming language (e.g., C, C#, etc.) or in any scientific computing environment (e.g., Matlab, Mathematica, etc.) The library is designed to be usable by any developer whether he is developing an image viewer and a GUI, doing batch processing, or even using the library in scientific research and pure scientific computing. Therefore, the functions in this library produce an output suitable for any application, leaving the interfacing issues to the application of interest. In particular for image viewer developers, they have to design the GUI suitable for rendering the output generated by LIBCAD . For Cloud-Based API, no advanced technical nor programming knowledge is required.

This document is not intended to be a piece of literature on the breast cancer detection methodologies. Therefore, the scientific background given below will be minimal and only sufficient for using LIBCAD. For interested readers, who want to get more details on designing breast cancer CAD, they may refer to Yousef et al. (2010) and other references cited in that work. For the purpose of this documentation we suffice with the following background.

In this Chapter, we provide such an introduction. Chapter 2 discusses software and hardware issues for using LIBCAD, e.g., dependencies, required packages, and installation. Chapter 4 is a detailed manual for LIBCADfunctions.

1.1 Technical Background For Breast Cancer Detection

"*Breast cancer is the most common cancer in women in developed Western countries (Althuis et al., 2005) and is becoming ever more significant in many developing countries (Yang et al., 2005). Although incidence rates are increasing, mortality rates are stable, representing an improved survival rate. This improvement can be attributed to effective means of early detection, mainly mammography, as well as to significant improvement in treatment options.*"(Freedman et al., 2006)

A Computer Aided Detection (or Computer-Assist Device) (CAD) "*refers to a computer algorithm that is used in combination with an imaging system as an aid to the human reader of images for the purpose of detecting or classifying disease*" (Wagner et al., 2002).

The American Food and Drug Administration (FDA) had approved the use of Computer Aided Devices (Detection or Diagnosis) in 1998; since then many CAD systems have been developed. Despite the availability of such systems all over the world, and in the U.S. in particular, they have no existence in many countries for their exaggerated price which ranges from 50,000$ to 175,000$ (Hall, 2007).

Detection of breast cancer while it is still small and confined to the breast provides the best chance of effective treatment for women with the disease (Ferlay et al., 2004; Parkin et al., 2005). Benefits of early detection include increased survival rate, increased treatment options and improved quality of life. Currently, there is insufficient knowledge about the causes of breast cancer for primary prevention strategies to reduce incidence in the population.

Causes of missed breast cancer on mammography can be secondary to many factors including those related to the patient (whether inherent or acquired), the nature of the malignant mass itself, poor mammographic techniques, or provider factors or interpretive skills of radiologists and oncologists, including perception and interpretation errors, (M. et al., 2007).

Perception error occurs when the lesion is included in the field of view and is evident but is not recognized by the radiologist. The lesion may or may not have subtle features of malignancy that cause it to be less visible. Small nonspiculated masses, areas of architectural distortion, asymmetry, small clusters of amorphous or faint microcalcifications may be difficult to perceive (M. et al., 2007).

Several factors may lead to misinterpretation, such as lack of experience, fatigue, or inattention. Misinterpretation may also occur if the radiologist fails to obtain all the views needed to assess the characteristics of a lesion or if the lesion is slow growing and prior images are not used for comparison (S., 2005; M. et al., 2007).

Many forms of abnormalities may appear in a mammogram. We focus here on two major types, masses and microcalcifications. *"A mass is a three-dimensional structure demonstrating convex outward borders, usually evident on two orthogonal views*" (ACR, 2003). See Figure B.2a for example.

Microcalcifications are "*deposits of calcium in the tissues. Calcification in the breast can be seen on a mammogram, but cannot be detected by touch. There are two types of breast calcification, macrocalcification and microcalcification. Macrocalcifications are large deposits and are usually not related to cancer. Microcalcifications are specks of calcium that may be found in an area of rapidly dividing cells. Many microcalcifications clustered together may be a sign of cancer.*" NCI (2011). See Figure B.3a

Breast lesions are tagged according to the different radiological lexicons and then categorized by the radiologist according to the "Breast Imaging Reporting and Data System" (BIRADS) scoring system. Table 1.1 is the description of the international BIRADS scoring system for diagnosis of breast lesions.

Category 0	mammographic assessment is incomplete
Category 1	negative
Category 2	benign finding(s)
Category 3	probably benign finding(s)
Category 4	suspicious abnormality
Category 5	highly suggestive of malignancy

Table 1.1: "Breast Imaging Reporting and Data System" (BIRADS) scoring.

The implementation of Computer aided detection (CAD) systems will help to reduce the human errors that lead to missing breast carcinoma, either related to poor perception or interpretation errors. CAD could increase the sensitivity of mammography interpretation (Muttarak et al., 2006). In retrospect, developing an affordable CAD library like LIBCAD for image viewer companies is valuable to the field of medical diagnosis.

1.2 Accuracy

1.2.1 Mass Detection

We consider a marked region as True Positive (TP) if its center falls in the marking of the radiologist. The image is considered to be TP if one of its mass regions is detected in that sense. False markers per image is the average number of markers produced on normal images.

The sensitivity of our current algorithm has been calculated across many databases come from different systems of digital mammography and from different institutions.

Database 1

false markers	0.2	0.85	1.5
TP Per Case	81.25%	90%	97%
TP Per Side	82.4%	91.2%	97%
TP Per Image	70%	91.25%	95%

Database 2

false markers	0.2	0.85	1.5
TP Per Case	78%	91.4 %	94%
TP Per Side	76.2%	90.5%	93%
TP Per Image	66.3%	87.25%	90 %

Database 3

false markers	0.2	0.85	1.5
TP Per Case	73%	89%	90%
TP Per Side	71.5%	87.5%	91.2%
TP Per Image	63.3%	81.7%	86.75%

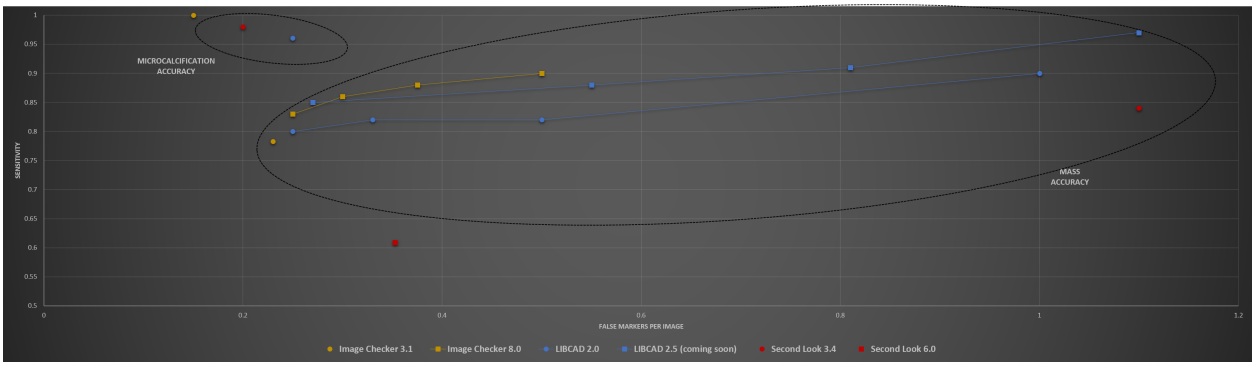
The sensitivity of the mass detection has been improved dramatically in the current version if compared to that of the first version (refer to Appendix A).

1.2.2 Microcalcification Detection

First, let's define a cluster of microcalcifications to be a set of detected foci, where any two of them are at most 3 mm apart. If the number of foci per cluster is lower than the selected threshold level, the cluster is considered undetected although those foci will be marked to the radiologist. A cluster is considered a Positive cluster, and hence is counted, if the number of its detected foci is larger than the selected threshold level and its center is located within the true marking (ground truth) of the radiologist proven by a biopsy. The performance is consistent across different databases, which we report at two different thresholds—6 and 4 foci per cluster—in the following table:

false markers	0.18	.38
TP Per Case	94%	96%
TP Per Side	94%	96%
TP Per Image	94%	96.6%

1.3 Comparative Accuracy



(a) Accuracy Comparison to commercial CADs

For details about the comparison, see:

1.3.1 Second Look

- Ver. 3.4 (CADx System) (Brem et al., 2005)
  - Mass: (84% per study TP), 1.1 FM; Micro: (98% per study TP), 0.2 FM.
- Ver. 6.0 (iCAD) (Ellis et al., 2007)
  - Mass: (60.9% per study, 42.6% per view TP), 1.41/4 FM, 1 – 31.4% FP.

1.3.2 ImageChecker (R2 Technology)

- Ver. 3.1 (Kim et al., 2010)
  - Mass: (78.3% per study, 65.2% per view TP), 0.23 FM; Micro: (100% per study, 94.0% per view TP), 0.15 FM.
- Ver. 8.0 (Roehrig, 2005)
  - Mass: (83.0% “seems” per study TP), 1.0/4 FM.
  - Mass: (86.0% “seems” per study TP), 1.2/4 FM.
  - Mass: (88.0% “seems” per study TP), 1.5/4 FM.
  - Mass: (90.0% “seems” per study TP), 2.0/4 FM.

CHAPTER 2

DEPENDENCIES, REQUIREMENTS, AND INSTALLATION

This Chapter lists the hardware and software requirements for using LIBCAD.

2.1 Software Requirements

- DLL: Supported Operating Systems is Microsoft Windows 7 64-bit.
- Shared Object: should be compatible with all Linux; however, only test on Ubuntu.
- Cloud-Based API: All Operating Systems are supported.

2.2 Hardware Requirements

No special hardware is currently required. However, the execution time of the library algorithms has been measured and tested on machines with Quad processors and 4GB of memory.

2.3 Software Installation

For DLL:

1. Make sure that you have Microsoft `.net` Framework 4.0 installed.
2. Install `Prerequisites\MCRInstaller.exe` : the MATLAB Run Time Library.
3. Install `Prerequisites\vcredist_x64.exe` : Microsoft Visual C++ 2010 Redistributable Package (x64).
4. Run the self extracting file `Setup.exe`; which you should run using the administrative privileges. (it will install LIBCAD and all required components to `C:\Program Files\LIBCAD` directory).

For Cloud-Based API:

1. A Supported Browser.

2.4 Image Requirements

We assume that the image to be passed to LIBCAD is a DICOM image produced by any digital mammography machine, with any gray-level or spatial resolution. It is worth mentioning that some mammography machines produce a non-contrasty images (see, e.g., Figure B.1a). No contrast adjusting is required for such an image, since contrasting and all image enhancements are part of the detection process.

CHAPTER 3

BACKGROUND INFORMATION

LIBCAD is a scientific library that was developed using Matlab[1]. Therefore, many data types used in this library are Matlab data types defined in `MWArray.dll`. This documentation is sufficient for efficient use of the library. However, if the reader is interested in more details about the used data types please refer to Mathworks (2011)[2]

3.1 Detecting Masses

Before explaining the functions, We give a brief description of how masses are detected in this library. For each pixel in the image, a score is assigned that corresponds to its likelihood of being abnormal. We call the resulting image "score image". Therefore, high scores correspond to abnormal regions and low scores correspond to normal ones. Viewing this score image is one way in which a radiologist can diagnose the case. Figure B.2 explains this more.

Figure B.2a shows the original image before detection (but after preprocessing, which is done automatically by the library). The image in Figure B.2b is the score image. It is clear that higher gray levels are associated with abnormal regions.

Producing the score image is provided by the function `FindMassesAndCentroids`. It also provides a list of centroids, each one is located in the center of a region that the LIBCAD considers to be high likely abnormal. The list is descendingly sorted based on the likelihood of a region to be abnormal.

It is worth mentioning that displaying the results (the score image or the centroids) is a matter of taste and depends on the application. If you are a company incorporating LIBCAD into your image viewer, you can consult your customer radiologists who will be using your viewer. You can adjust the interface for their convenience. The following is a set of suggested displaying options.

- Displaying **score image** in a separate window, side-by-side to the original image (Figure B.2a–B.2b).
- Displaying **centroids** of the detected regions overlayed on the original image (Figure B.2c).

3.2 Detecting Microcalcifications

Microcalcifications are very small deposits which appear in the mammogram as tiny dots. The indices of the detected foci are returned by the function FindMicrocalcifications. We define microcalcification cluster as follows: It is a set of detected foci which are, at most, 25 pixels apart. The number of foci per cluster is determined by a threshold value.

Figure B.3a shows an image of clustered microcalcifications, figure B.3b shows the same image with the detected foci highlighted. Each microcalcification focus is represented by its two coordinates. Figure B.4b and figure

[1] Matlab is a product of The MathWorks, Inc., `www.MathWorks.com`

[2] Go to the link and, first, create an account.

B.4c show how tuning the threshold (number of foci per microcalcification cluster) affects the resulted score image of the original image in figure B.4a.

3.3 Image Enhancement

Some of mass and microcalcification regions may be very unclear and subtle according to its shape, surrounding tissues and the density of the breast and in this case, unfortunately, the radiologist is very probable to fail to detect the abnormality even with the help of LIBCAD; therefore, enhancing the image is desired. Image enhancement is the process of removing artifacts, reducing the density of the breast, stripping the less interesting details off, and further exposing the details of abnormal region(s). Enhancing the image could be done either by enhancing the entire image at once, or by enhancing a portion of the image by a window.

Enhancing the entire image is very useful in assisting the radiologist to detect abnormalities, if any, in fuzzy cases, as it reduces significantly the density of the breast; consequently, facilitating the task of diagnosing the case. In addition, enhancing a part of the image at a time is useful in cases such as determining the status of a specific controversial region and better understanding of unusual pattern.

Figure B.5a shows an image before any further enhancement, while figure B.5b shows the image after it was entirely enhanced, and figure B.5c shows the image after enhancing a portion of it.

CHAPTER 4

DLL AND SHARED OBJECT LIBRARY

This Chapter is a detailed programmer guide for using LIBCAD. We explain what every function does, along with its input and output parameters. A complete C# project, found on the installation DVD, is provided as an example for using the whole library.

4.1 Calling LIBCAD

1. Make sure you have followed the installation procedure in Section 2.3 and LIBCAD is installed in `C:\Program Files\LIBCAD`.
2. Add the following references to the C# project (The procedure of adding references depends on the version of your Visual Studio) [1]
   - `CAD.dll`
   - `MWArray.dll`
3. Add `CAD.CTF` file as an existing item to your project [2] and set its 'Copy to output directory' property to 'Copy Always'.
4. Add this line to your code:

```
using MathWorks.MATLAB.NET.Arrays;
```

5. Finally, instantiate an object of the CAD Class; e.g.,

```
CAD.CAD libcad = new CAD.CAD();
```

6. After deploying your software on the client's machine, you must run it using the administrative privileges.

[1] For 2010 version, visit this link: `"http://msdn.microsoft.com/en-us/library/wkze6zky(v=vs.100).aspx"`
[2] For 2010 version, visit this link:`"http://msdn.microsoft.com/en-us/library/9f4t9t92(v=vs.100).aspx"`

4.2 Function Documentation

4.2.1 Detection tools

FindMassesAndCentroids

Function Interface:

```
public MWArray[] FindMassesAndCentroids(int numArgsOut, MWArray imagePath);
```

Calling Example:

```
MWArray    imagePath  = "c:\\MyDirectory\\I0";
MWArray[]  outputs    = libcad.FindMassesAndCentroids(2, imagePath);
MWArray    scoreImage = outputs[0];
MWArray    centroids  = outputs[1];
```

Parameters:

```
/*
* numArgsOut: A number expressing how to access the output array (which is of type MWArray). The
              most convenient way is to pass its length 2 in this case); for details see the
              MWArray manual in the References.
* imagePath : Full path name of the DICOM image.
* outputs[0]: Score image.
* outputs[1]: n by 3 array holds the coordinates of the centroids represent the masses along with
              its probabilities (The array is sorted descendingly according to the probabilities).
*/
```

Description:

This function takes the full path name of a DICOM image and returns a score image (`outputs[0]`), along with the coordinates of all centroids represent the masses and the probability of each centroid (`outputs[1]`).

The score image (`outputs[0]`) expresses the likelihood of each pixel to be abnormal. It can be displayed directly, without any processing, to aid radiologist in diagnosis and detecting abnormal regions.

The 2D array (`outputs[1]`) contains the coordinates of the centroids that represent the masses and the probability of each centroid. It is another way to help the radiologist to detect the masses by displaying a small mark in the center of the region which has high probability of being abnormal.

Hint: It is worth mentioning that the function `FindMassesAndCentroids` is fast as it takes about 10 seconds per image.

FindMicrocalcifications

Function Interface:

```
public MWArray[] FindMicrocalcifications(int numArgsOut, MWArray imagePath);
```

Calling Example:

```
MWArray     imagePath  = "c:\\MyDirectory\\I0";
MWArray[]   outputs    = libcad.FindMicrocalcifications(1, imagePath);
MWArray     positions  = outputs[0];
```

Parameters:

```
/*

* numArgsOut: a number expressing how to access the output array (which is of type MWArray). The
              most convenient way is to pass its length (1 in this case); for details see the
              MWArray manual in the References.
* imagePath : full path name of the DICOM image.
* outputs[0]: a 2-dimensional array that holds the indices of all microcalcifications.
*/
```

Description:

This function detects any microcalcification (which is a small and tiny deposit), whether it is clustered or not. It takes the full path name of a DICOM image (`imagePath`) and returns the indices of the foci (`outputs[0]`). Each microcalcification focus is represented by its two coordinates.

The detected microcalcifications are expressed by the indices (`outputs[0]`). It could be displayed directly, without any further processing, to aid radiologist in diagnosis. You can also overlay it on the original image with different transparent color.

Hint: It is worth mentioning that the function `FindMicrocalcifications` is fast as it takes about 3 seconds per image.

4.2.2 Enhancement tools

LoadImage

Function Interface:

```
public MWArray[] LoadImage(int numArgsOut, MWArray imagePath);
```

Calling Example:

```
MWArray     imagePath    = "c:\\MyDirectory\\I0";
MWArray[]   outputs      = libcad.LoadImage(2, imagePath);
MWArray     image        = outputs[0];
MWArray     enhanceParam = outputs[1];
```

Parameters:

```
/*
* numArgsOut: a number expressing how to access the output array (which is of type MWArray). The
              most convenient way is to pass its length (2 in this case); for details see the
              MWArray manual in the References.
* imagePath : full path name of the DICOM image.
* outputs[0]: a 2-dimensional array contains the DICOM image.
* outputs[1]: a 1-dimensional array contains parameters needed for the process of image
              enhancement.
*/
```

Description:

This function loads the DICOM image in the memory. It takes the full path name of a DICOM image (`imagePath`) and returns the DICOM image (`outputs[0]`) in a 2-dimensional array. along the DICOM image, the function returns an array contains parameters that are necessary for the enhancement process (`outputs[1]`).

EnhanceImage

Function Interface:

```
public MWArray[] EnhanceImage(int numArgsOut, MWArray image, MWArray enhanceParam, MWArray
                enhancementFactor);
```

Calling Example:

```
MWArray      imagePath           = "c:\\MyDirectory\\I0";
MWArray[]    loadImageOutputs    = libcad.LoadImage(2, imagePath);
MWArray      image               = loadImageOutputs[0];
MWArray      enhanceParam        = loadImageOutputs[1];
MWArray      enhancementFactor   = 10;
MWArray[]    enhanceImageOutputs = libcad.EnhanceImage(1,image, enhanceParam, enhancementFactor);
MWArray      enhancedImage       = enhanceImageOutputs[0];
```

Parameters:

```
/*
* numArgsOut:              number expressing how to access the output array (which is of type
              MWArray). The most convenient way is to pass its length (1 in this case); for
              details see the MWArray manual in the References.
* image:                   2-dimensional array contains the DICOM image.
* enhanceParam:            1-dimensional array contains parameters needed for enhancing the image.
* enhancementFactor:       number which is tuning parameter of the enhancement.
* enhanceImageOutputs[0]: 2-dimensional array holds the enhanced DICOM image.
*/
```

Description:

This function takes the DICOM image that was loaded in memory by LoadImage function. Then, it applies the enhancement according to the passed arguments (`enhanceParam`) and (`enhancementFactor`) and finally it returns the enhanced image to be shown to the radiologist. Enhancement is very useful in revealing the hidden details to the radiologist and moreover discarding artifacts as well as less interesting details. Therefore, decreases false alarms and increases dramatically the overall performance of the radiologist.

LocalEnhanceImage

Function Interface:

```
public MWArray[] LocalEnhanceImage(int numArgsOut, MWArray image, MWArray enhanceParam, MWArray
            enhancementFactor, MWArray x, MWArray y, MWArray level);
```

Calling Example:

```
MWArray      imagePath                 = "c:\\MyDirectory\\I0";
MWArray[]    loadImageOutputs          = libcad.LoadImage(2, imagePath);
MWArray      image                     = loadImageOutputs[0];
MWArray      enhanceParam              = loadImageOutputs[1];
MWArray      enhancementFactor         = 10;
MWArray      x                         = 100;
MWArray      y                         = 100;
MWArray      level                     = 5;
MWArray[]    localEnhanceImageOutputs = libcad.LocalEnhanceImage(1, image, enhanceParam,
             enhancementFactor, x, y, level);
MWArray      enhancedImage             = localEnhanceImageOutputs[0];
```

Parameters:

```
/*

* numArgsOut:                number expressing how to access the output array (which is of type
            MWArray). The most convenient way is to pass its length (1 in this case); for
            details see the MWArray manual in the References.
* image:                     2-dimensional array contains the DICOM image.
* enhanceParam:              1-dimensional array holds parameters needed for the enhancing.
* enhancementFactor:         number which is tuning parameter of the enhancement.
* x:                         number which is the horizontal coordinate of the point in the center
             of the window which will be enhanced.
* y:                         number which is the vertical coordinate of the point in the center
            of the window which will be enhanced.
* level:                     number of levels around the point in the center of the window. (i.e.
            window width = (2*level) + 1 e.g. if the level is 3, so it means the width of the
            window is 3+1+3 = 7)
* localEnhanceImageOutputs[0]: 2-dimensional array holds the enhanced DICOM image.
*/
```

Description:

This function performs the same operation done by the previous function but only on a specific area on the image defined by a window. that window in turn is determined by the coordinates of the point in its center and the number of levels around it. Local enhancement is useful in revealing the hidden details in an interesting area to the radiologist and therefore decreases false alarms.

CHAPTER 5

CLOUD-BASED API

This Chapter is a detailed guide for using LIBCAD. We explain what every function does, along with its input and output parameters.

5.1 Function Documentation

5.1.1 Detection tools

FindMassesAndCentroids

Inputs:

```
* image: Binary DICOM image with ID "image".
```

Outputs:

```
*  $output : Encoded JSON string consists of the following keys :
*  downloadService : A service which downloads the output image directly when getting on it.
*  imageWidth : The output image width.
*  imageHeight : The output image height.
*  centroids : Array of objects represents set of detected masses centroids.
*  x: Double represents (x) position of a centroid on the image.
*  y: Double represents (y) position of a centroid on the image.
*  p: Double represents the probability of the malegnancy for centroids.
```

Description:

This function takes a DICOM image and returns a JSON string (`outputs`), containing keys that represnt: output image, size of the output image, and coordinates of all centroids represent the masses and the probability of each centroid.

The output image, which is downloaded by (`downloadservice`), expresses the likelihood of each pixel to be abnormal. It can be displayed directly, without any processing, to aid radiologist in diagnosis and detecting abnormal regions.

Hint: It is worth mentioning that the function `FindMassesAndCentroids` is fast as it takes about 10 seconds per image.

FindMicrocalcifications

Inputs:

```
* image: Binary DICOM image with ID "image".
```

Outputs:

```
*  $output : Encoded JSON string consists of the following keys :
*  positions : Array of objects represents set of detected microcalcifications
*  x: Double represents (x) position of a microcalcification on the image.
*  y: Double represents (y) position of a microcalcification on the image.
```

Description:

This function detects any microcalcification (which is a small and tiny deposit), whether it is clustered or not. The function takes the original DICOM image (`image`) and returns an encoded JSON string (`outputs`), containing keys that represnt: coordinates of all Microcalcifications found in the image. Each microcalcification focus is represented by its two coordinates.

The detected microcalcifications are expressed by the array (`positions`). It could be displayed directly, without any further processing, to aid radiologist in diagnosis. You can also overlay it on the original image with different transparent color.

Hint: It is worth mentioning that the function `FindMicrocalcifications` is fast as it takes about 3 seconds per image.

5.1.2 Enhancement tools

EnhanceImage

Inputs:

```
*  image              : Binary DICOM image with ID "image".
*  enhancementFactor  : Integer represents enhancement tuning factor. usually around 10.
```

Outputs:

```
*  $output             : Encoded JSON string consists of the following keys :
*  downloadService : A service which downloads the output image directly when getting on it.
*  imageWidth          : The output image width.
*  imageHeight         : The output image height.
```

Description:

This function takes the source DICOM image. Then, it applies the enhancement according to the passed argument (`enhancementFactor`) and finally it returns the enhanced image to be shown to the radiologist. Enhancement is very useful in revealing the hidden details to the radiologist and moreover discarding artifacts as well as less interesting details. Therefore, decreases false alarms and increases dramatically the overall performance of the radiologist.

LocalEnhanceImage

Inputs:

```
*  image                 :Binary DICOM image with ID "image".
*  enhancementFactor     : Integer represents enhancement tuning factor. usually around 10.
*  x                     : number represents the horizontal coordinate of the point in the center
            of the window which will be enhanced .
*  y                     : number represents the vertical coordinate of the point in the center of
            the window which will be enhanced .
*  level                 : double represents the width of the square partial view to be enhanced .
```

Outputs:

```
*  $output[0]          : JSON encoded string consists of the following keys :

*  positions           : An array that holds the indicies of all microcalcifications.
```

Description: This function performs the same operation done by the previous function but only on a specific area on the image defined by a window. that window in turn is determined by the coordinates of the point in its center and the number of levels around it. Local enhancement is useful in revealing the hidden details in an interesting area to the radiologist and therefore decreases false alarms.

CHAPTER 6

FAQ

- Hint: You can run visual studio as administrator, so you can debug, however know that FindMasses will **NOT** work in visual studio debugging mode.
- Hint: If you get the exception "Method Invokation Failed", make sure the architecture of the DLL is the same as that used in the project build, i.e. if the DLL is of architecture x64 for example, do not use it in a project of build x32 or Mixed Platform.

APPENDIX A

JUST FOR HISTORY: SENSITIVITY OF MASS DETECTION OF VER. 1 (CURRENTLY OBSOLETE)

Database 1

false markers	0.2	0.85	1.5
TP Per Case	20%	60%	78%
TP Per Side	20%	60%	80%
TP Per Image	10%	42%	63%

Database 2

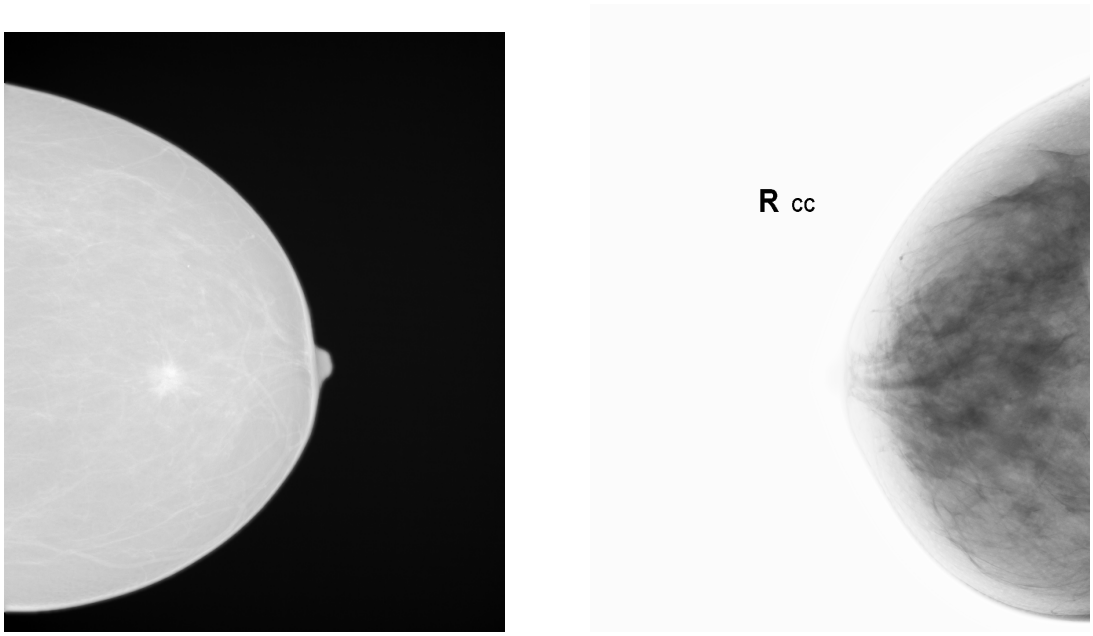
false markers	0.2	0.85	1.5
TP Per Case	58%	85 %	91%
TP Per Side	58%	86%	92%
TP Per Image	44%	70%	77 %

Database 3

false markers	0.2	0.85	1.5
TP Per Case	45%	78%	84%
TP Per Side	45%	77%	82%
TP Per Image	35%	67%	71%

APPENDIX B

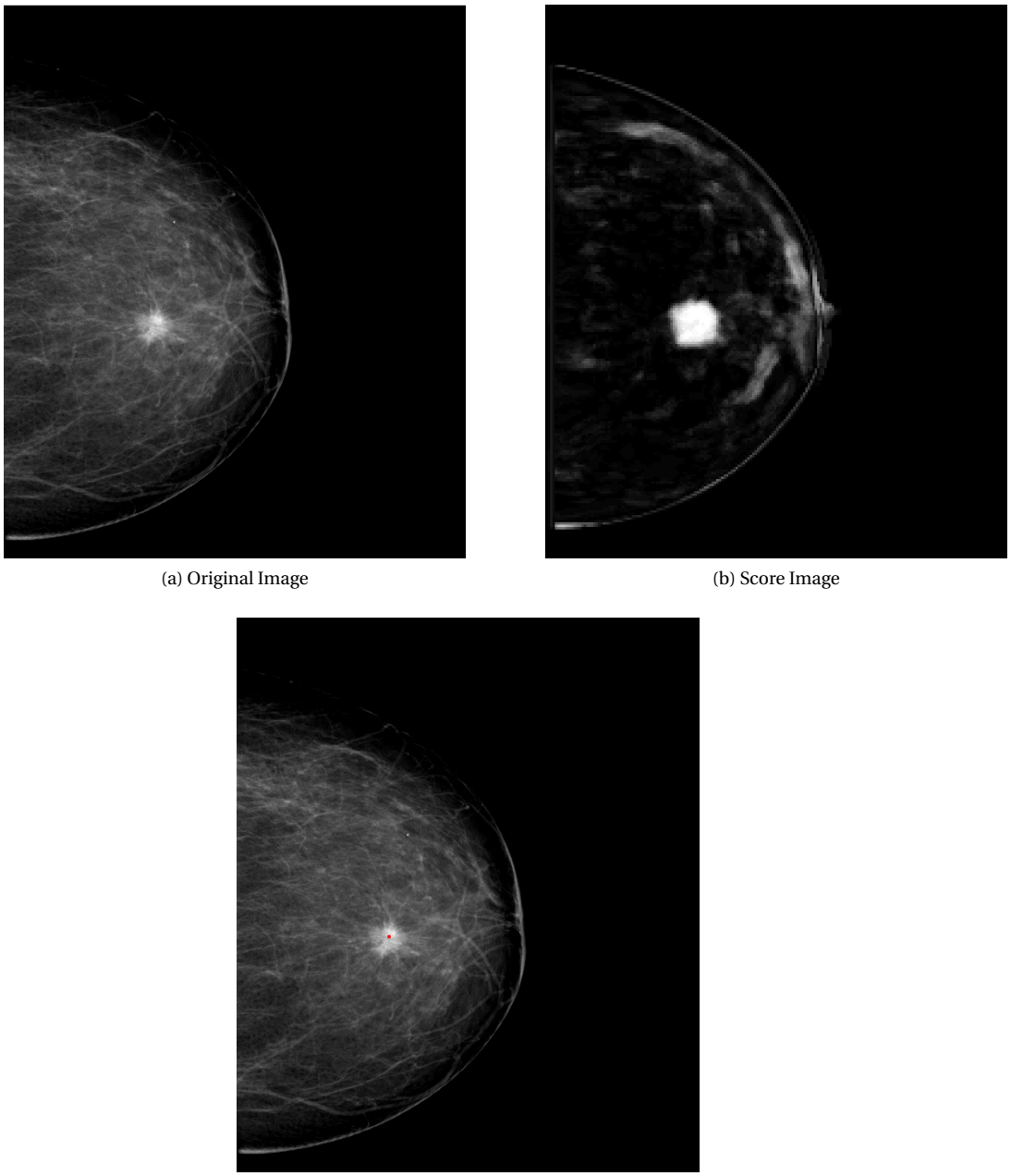
IMAGE EXAMPLE



(a) Non-contrasty image

(b) White-background image with label

Figure B.1: Typical digital mammograms without preprocessing.

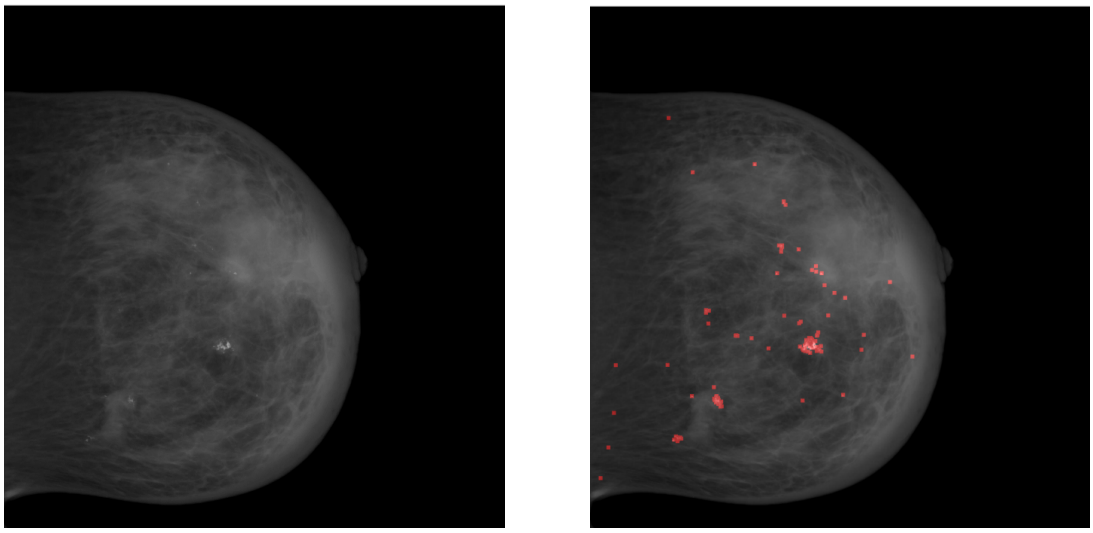


(a) Original Image

(b) Score Image

(c) A centroid represented as a small mark overlayed on the original image.

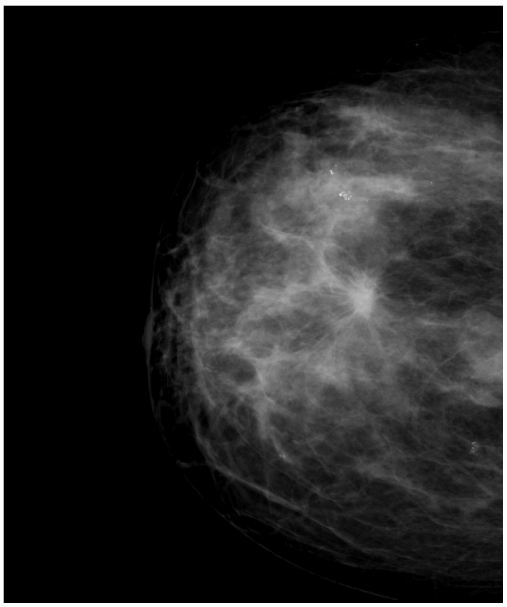
Figure B.2: Original image and its score image. Pixels in the score image have real values (`double`).



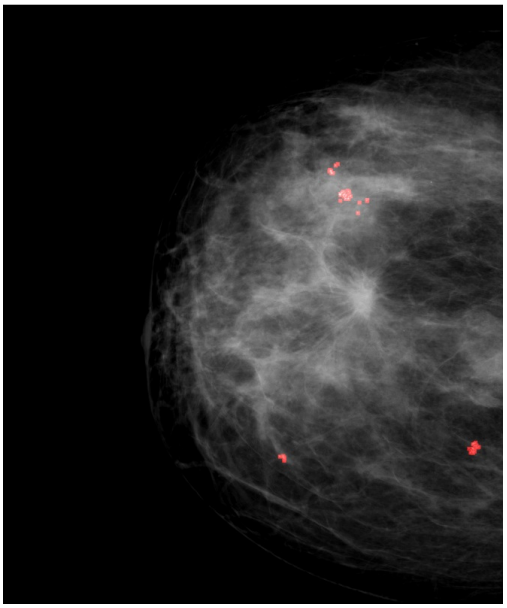
(a) Original Image

(b) Score Image

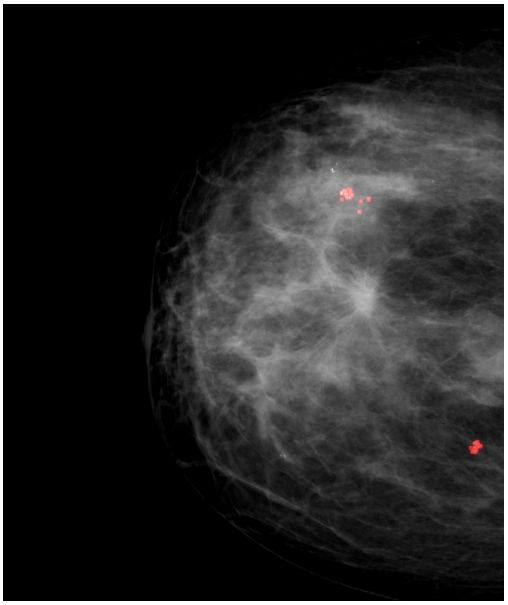
Figure B.3: The original image and the detected foci indices highlighted and overlayed on the original image respectively.



(a) Original microcalcification image.

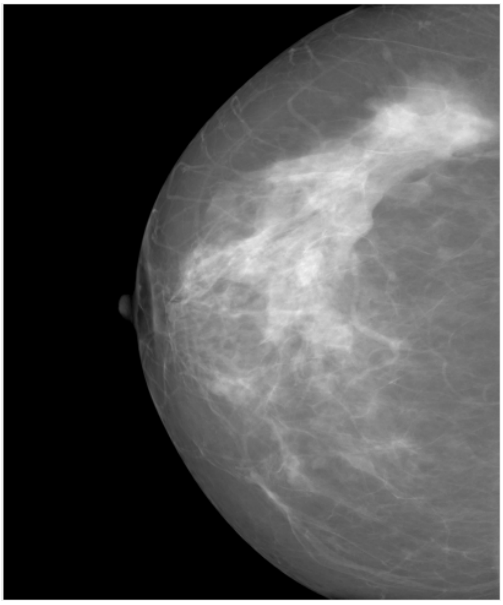


(b) Score image thresholded at θ_1 and overlayed (transparent red color) on the original image.

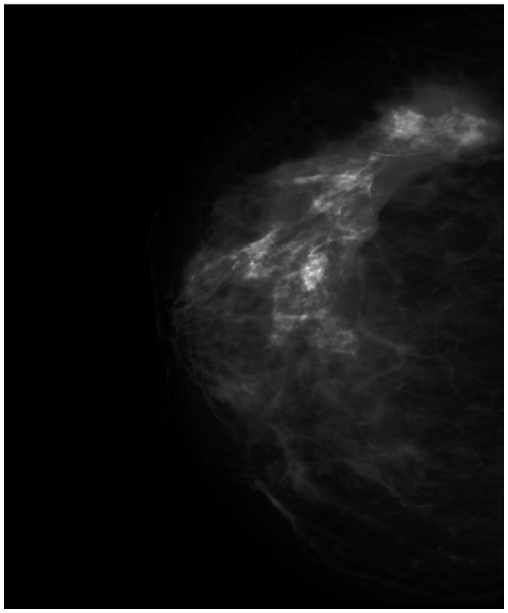


(c) Score image thresholded at $\theta_2 > \theta_1$ (less aggressive) and overlayed (transparent red color) on the original image.

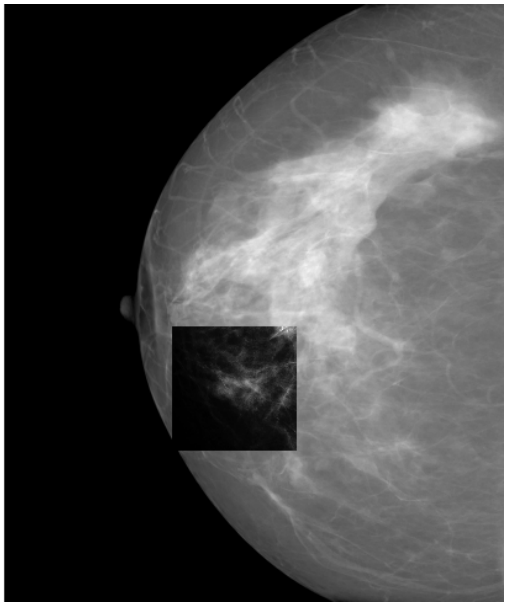
Figure B.4



(a) Original image



(b) Image after entirely enhanced.



(c) Portion of the image is enhanced.

Figure B.5

BIBLIOGRAPHY

ACR, 2003. Breast Imaging Reporting and Data System Atlas (BI-RADS Atlas), 4th Edition. American Colleage of Radiology, Reston, VA.

Althuis, M. D., Dozier, J. M., Anderson, W. F., Devesa, S. S., Brinton, L. A., 2005. Global trends in breast cancer incidence and mortality 1973-1997. Int. J. Epidemiol. 34 (2), 405–412.
URL http://ije.oxfordjournals.org/cgi/content/abstract/34/2/405

Brem, R. F., Hoffmeister, J. W., Zisman, G., DeSimio, M. P., Rogers, S. K., 2005. A computer-aided detection system for the evaluation of breast cancer by mammographic appearance and lesion size. AJR Am J Roentgenol 184 (3), 893–896.
URL http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=15728614

Ellis, R. L., Meade, A. A., Mathiason, M. A., Willison, K. M., Logan-Young, W., 2007. Evaluation of Computer-aided Detection Systems in the Detection of Small Invasive Breast Carcinoma1. Radiology 245 (1), 88–94.
URL http://radiology.rsna.org/content/245/1/88.abstract

Ferlay, J., Bray, F., Pisani, P., Parkin, D. M., 2004. GLOBOCAN 2002: Cancer Incidence, Mortality and Prevalence Worldwide.

Freedman, L. S., Edwards, B. K., Ries, L. A. G., Young, J. L., 2006. Cancer Incidence in Four Member Countries (Cyprus, Egypt, Israel, and Jordan) of the Middle East Cancer Consortium (MECC) Compared with US SEER.

Hall, F. M., 2007. Breast Imaging and Computer-Aided Detection. N Engl J Med 356 (14), 1464–1466.
URL http://content.nejm.org

Kim, S. J., Moon, W. K., Kim, S. Y., Chang, J. M., Kim, S. M., Cho, N., 2010. Comparison of two software versions of a commercially available computer-aided detection (CAD) system for detecting breast cancer. Acta Radiol 51 (5), 482–490.
URL http://www.ncbi.nlm.nih.gov/pubmed/20429765

M., K. R., M., A. R. N., A., H. M., A., S. M., 2007. Missed Breast Carcinoma; Why and How to Avoid? Journal of the Egyptian National Cancer Institute Volume 19 (3).

Mathworks, 2011. Dotnet builder documentation.
URL http://www.mathworks.com/help/toolbox/dotnetbuilder/MWArrayAPI/HTML/index.html

Muttarak, M., Pojchamarnwiputh, S., Chaiwun, B., 2006. Breast carcinomas: why are they missed? Singapore Med J 47 (10), 851–857.
URL http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=16990959

NCI, 2011. Dictionary of cancer terms.
URL http://www.cancer.gov/dictionary

Parkin, D. M., Bray, F., Ferlay, J., Pisani, P., 2005. Global cancer statistics, 2002. Ca-a Cancer Journal for Clinicians 55 (2), 74–108.
URL <GotoISI>://000231484000005http://caonline.amcancersoc.org/cgi/reprint/55/2/74.pdf

Roehrig, J., 2005. The manufacturer's perspective. Br J Radiol 78 Spec No, S41–5.
URL http://www.ncbi.nlm.nih.gov/pubmed/15917445http://bjr.birjournals.org/cgi/reprint/78/suppl_1/S41.pdf

S., R., 2005. Imaging Techniques Alternative to Mammography for Early Detection of Breast cancer. TRCT, 1–120.

Wagner, R. F., Beiden, S. V., Campbell, G., Metz, C. E., Sacks, W. M., 2002. Assessment of Medical Imaging and Computer-assist Systems: Lesson from Recent Experience. Acad Radiol 9, 1264–1277.

Yang, L., Parkin, D. M., Ferlay, J., Li, L., Chen, Y., 2005. Estimates of Cancer Incidence in China for 2000 and Projections for 2005. Cancer Epidemiol Biomarkers Prev 14 (1), 243–250.
URL http://cebp.aacrjournals.org/cgi/content/abstract/14/1/243

Yousef, W. A., Mustafa, W. A., Ali, A. A., AbdelRazek, N. M., Farrag, A. M., 2010. On Detecting Abnormalities in Digital Mammography. In: Applied Imagery Pattern Recognition Workshop, 2010. Proceedings. 39rd; IEEE Computer Society.