

ARTICLE

IMAGE AND VLSI TECHNIQUES DEVELOPMENT OF EARLY DETECTION OF BREAST CANCER AND DIAGNOSE USING MAMMOGRAMS

B Thamilvalluvan¹, M Anto Bennet^{2*}, Priyanka Patee Alphonse³, K Sujithra⁴, Soda Chandrasekhar⁵, C Kannan⁶, D R Thendralarasi⁷

¹Professor, ²Asst.professor, ^{3,4,5,6,7}UG Students, Department of ECE, VEL TECH ,Avadi, Chennai 600 062, Tamil Nadu, INDIA

ABSTRACT

In this paper we proposed a method to detect the breast cancer accurately in the early stage using image processing and VLSI techniques. In the existing methods cancer affected regions of the mammogram were detected but could not calculate the affected number of cells in that region. In our proposed method a new algorithm has been developed which is able to find the cancer affected regions in the mammogram very accurately and also find the number of cells in the affected region. Initially the mammogram is taken as an input and was read on MATLAB. This mammogram is then converted into a text file based on the gray levels of the mammogram. Then the text file is taken as input in the Xilinx. Using verilog code text file is read and the file is assigned to a variable in binary format. The binary values of the text is read by using the verilog code and the binary values of the text is then compared with the default binary value that is been taken as a binary value of the cancer cell in the verilog code. Then by converting the binary values of the text into decimal value, the cancer cells are detected. Then output of the Xilinx is obtained in the form of waveforms and a text file is created by the output waveforms. Then the decimal values that match with the default decimal values are detected as cancer cells. Finally this text is further converted into image using MATLAB and the number of cancer cells calculated so as to find the affected area.

INTRODUCTION

Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modelled in the form of multidimensional systems. The identification of objects in an image and this process would probably start with image processing techniques such as noise removal, followed by (low-level) feature extraction to locate lines, regions and possibly areas with certain textures. The clever bit is to interpret collections of these shapes as single objects, e.g. cars on a road, boxes on a conveyor belt or cancerous cells on a microscope slide. One reason this is an AI problem is that an object can appear very different when viewed from different angles or under different lighting. Another problem is deciding what features belong to what object and which are background or shadows etc. The human visual system performs these tasks mostly unconsciously but a computer requires skilful programming and lots of processing power to approach human performance. Manipulation of data in the form of an image through several possible techniques. An image is usually interpreted as a two-dimensional array of brightness values, and is most familiarly represented by such patterns as those of a photographic print, slide, television screen, or movie screen [1-4]. An image can be processed optically or digitally with a computer. To digitally process an image, it is first necessary to reduce the image to a series of numbers that can be manipulated by the computer. Each number representing the brightness value of the image at a particular location is called a picture element, or pixel. A typical digitized image may have 512 × 512 or roughly 250,000 pixels, although much larger images are becoming common. Once the image has been digitized, there are three basic operations that can be performed on it in the computer. For a point operation, a pixel value in the output image depends on a single pixel value in the input image. For local operations, several neighboring pixels in the input image determine the value of an output image pixel. In a global operation, all of the input image pixels contribute to an output image pixel value. One of the techniques in image processing is image segmentation. The inevitable need for image segmentation is to analyze the images in precise manner. It is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics. Segmentation is a collection of methods allowing interpreting spatially close parts of the image as objects. Regions (i.e., compact sets) represent spatial closeness naturally and thus are important building steps towards segmentation. The object is everything what is of interest in the image (from the particular Application point of view). The rest of the image is background. The approach is similar to that used in pattern recognition, i.e., division of the image into set of equivalence classes. The purpose of image segmentation is to partition an image into meaningful regions with respect to a particular application. The segmentation is based on measurements taken from the image and might be grey level, colour, texture, depth or motion [5, 6, 7].

KEY WORDS

Transductive Support Vector Machine (TSVM), Many Textural Elements (TEXEL), Local Binary Pattern (LBP).

Received: 24 October 2016
Accepted: 20 December 2016
Published: 15 February 2017

*Corresponding Author
Email
bennetmab@gmail.com

MATERIALS AND METHODS

In this paper we proposed a method to detect the breast cancer accurately using image processing and VLSI techniques. Initially the mammogram is taken as an input and was read on MATLAB. This mammogram is then converted into a text file based on the gray levels of the mammogram. Then the text file is taken as input in the Xilinx. Using verilog code text file is read and the file is assigned to a variable in binary format. The binary values of the text is read by using the verilog code and the binary values of the text is then compared with the default binary value that is been taken as a binary value of the cancer cell in the verilog code. Then by converting the binary values of the text into decimal value, the cancer cells are detected. Then output of the Xilinx is obtained in the form of waveforms and a text file is created by the output waveforms. Then the decimal values that match with the default decimal values are detected as cancer cells. Finally this text is further converted into image using MATLAB and the number of cancer cells is calculated so as to find the affected area.

SIGNS OF CANCER IN A MAMMOGRAM

There are several signs of cancer to look for in a mammogram. The primary signs are local distortion of glandular tissue and existents of malignant micro calcifications. These signs may appear alone or together. Skin thickening, skin distortion, retraction of the nipple is considered as secondary signs. The attenuation of tumor may vary from depending upon type of tumor. Benign tumors are usually rounded and have distinct border. While malignant tumors tend to be more irregularly in shape, often speculated and have diffused borders. Normally two breast of one woman are very alike , and cancer of both breast is relatively rare .These facts help radiologist in the search of cancer in a mammogram. Micro calcification is microscopic grains of calcium produced be the cells as the result of some benign or malignant process. Most calcification is the result of some benign process. The may for instance be the rest of broken down cells, a cyst or milk. Benign tumors & malignant calcification differs in shape density and distribution However, it may be found that the predictions possible from the available information are limited because patients with very similar values on all the factors have greatly varying times to relapse. While this problem can be identified by analysis it cannot be addressed by it—the reason for it may be that the risk of relapse depends on some unmeasured factor or is simply subject to great random variation. From a statistical viewpoint, an important feature of the data is the presence of censoring, so that some of the times to relapse are not known exactly, only that they are greater than a certain value. This occurs either because follow-up is no longer available on the patient for some reason (maybe they have died of an unrelated cause), or simply because no relapse has yet occurred, when the survival time is known only to be as long as the total time elapsed since surgery. We assume that censoring times are essentially non-informative, so that for example censoring does not routinely occur just before a relapse is shown in [Fig. 1].

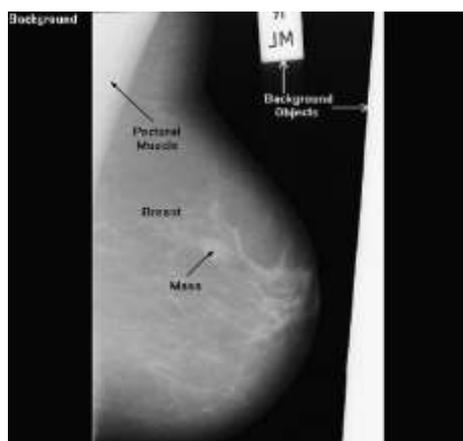


Fig. 1: Illustration of components of mammogram

Texture analysis is widely used for computer vision and image processing for classification and segmentation of image based on a local spatial variation of intensity or color. Existing texture analysis can be classified into statistical and structural methods. Statistical approach computes different properties are suitable if texture primitive sizes are comparable with the pixel sizes. In structural texture analysis method, the texture region is defined to have a constant texture if a set of local statistics or other local properties of the image are constant, slowly varying or approximately periodic. Majority of the existing texture analysis assumes that the texture images are acquired from the same viewpoint (same scale and same orientation). This gives a limitation of these methods. In many practical applications, it is very difficult or impossible to

ensure that images captured have the same translations, rotations or scaling between each other. Texture analysis should be ideally invariant to viewpoint. But recently model based method is also employed which includes autoregressive model, Gaussian Markov random fields, Gibbs random fields, World medial, wavelet model, multichannel Gabor model and steerable pyramid, etc. These models provide more powerful tools for invariant texture analysis. In statistical methods, texture is described by a collection of statistics of selected features. The statistics are broadly classified into first-order statistics, second-order statistics, and higher-order statistics. In structural methods, texture is viewed as consisting of many textural elements (called Texel) arranged according to some placement rules. In model based a texture image is modelled as a probability model or as a linear combination of a set of basic functions. The coefficients of these models are used to characterize texture images.

GREY LEVEL RUN LENGTH MATRIC

Gray level run length matrix is a statistical texture descriptor. This method involves the counting the number of pixels that have the same intensity in a particular direction. Run length is the number of adjacent pixels that have same gray level intensity in a particular direction shown in [Fig. 2]. The texture feature extracted from run length matrices produces great classification results. The texture features that can be extracted by GLRLM includes Short Run Emphasis (SRE), Long Run Emphasis (LRE), Low gray level Run Emphasis (LGLRE), High Gray Level Run Emphasis (HGLRE), Short Run Low Gray Level Emphasis (SRLGLE), Short Run High Gray level Emphasis (SRHGLE), Long Run Low Gray Level Emphasis (LRLGLE), Long Run High gray Level Emphasis (LRHGLE), Gray Level No uniformity (GLN), Run length no uniformity (RLN), Run Percentage (RPERC).

LOCAL BINARY PATTERN OPERATOR

LBP operator combines the characteristics of statistical and structural texture analysis. The LBP operator is used to perform gray scale invariant two-dimensional texture analysis. The LPB operator labels the pixel of an image by thresholding the neighbourhood (i.e. 3×3) of each pixel with the centre value and considering the result of this thresholding as a binary number. When all the pixels have been labelled with the corresponding LBP codes, histogram of the labels are computed and used as a texture descriptor. Given a pixel in the image LBP code can be computed by comparing it with its neighbours K is the maximal LBP pattern value. The most important properties of LBP features are computational simplicity and tolerance against the monotonic illumination changes. The basic LBP operator cannot be used for the dominant features of large scale structures. So it is extended to facilitate the analysis of textures with multiple scales by combining neighbourhoods with different sizes.

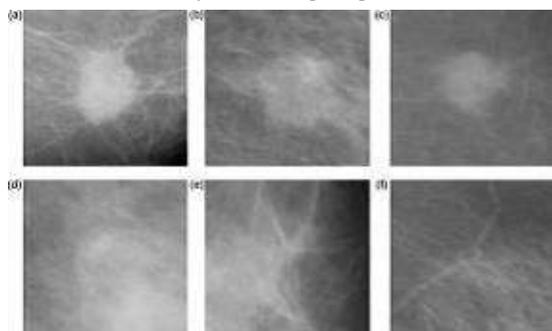


Fig. 2: (a-c) ROI with masses (d-f) ROI without masses

GREY-LEVEL DIFFERENCE STATISTICS

The Gray level difference method is based on the histogram of absolute difference between pairs of gray level. Several texture measure like mean, standard deviation, entropy; contrast can be obtained from Gray level Difference Statistics. This is shown in [Fig. 3].



Fig. 3: Texture measure from Gray level difference statistics.

.....
FLOW DIAGRAM

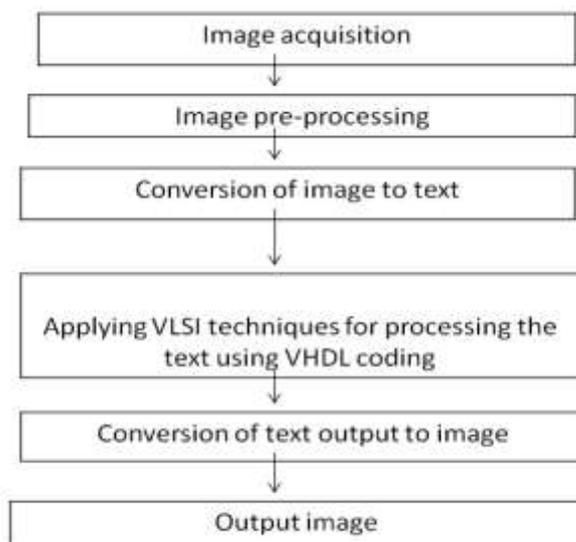


Fig. 4: Block Diagram of the proposed system

.....
IMAGE ACQUISITION

In [Fig. 4] ,Image acquisition in image processing can be broadly defined as the action of retrieving an image from some source, usually a hardware-based source, so it can be passed through whatever processes need to occur afterward. Performing image acquisition in image processing is always the first step in the workflow sequence because, without an image, no processing is possible. The image that is acquired is completely unprocessed and is the result of whatever hardware was used to generate it, which can be very important in some fields to have a consistent baseline from which to work. One of the ultimate goals of this process is to have a source of input that operates within such controlled and measured guidelines that the same image can, if necessary, be nearly perfectly.

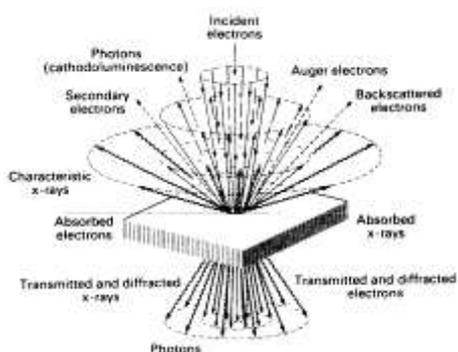


Fig. 5: signals generated by the interaction of an electron beam with a solid reproduced under the same conditions so anomalous factors are easier to locate and eliminate.

.....
IMAGE PRE-PROCESSING

Image pre-processing can significantly increase the reliability of an optical inspection. Several filter operations which intensify or reduce certain image details enable an easier or faster evaluation. Users are able to optimize a camera image with just a few clicks.

CONVERSION OF IMAGE TO TEXT FILE

An image is a 2D matrix. For converting into text file, it needs to be converted into 1D matrix and then to text file using file write in matlab. The image size is also a constraint, so it needs to be fixed for conversion. Then converted to 1D matrix using reshape command. Then write into text file.

```
A=imread('1.bmp');  
B=rgb2gray(A);  
disp('Image file read successful');  
%Figure,imshow(B),title ('org');  
C=imresize(B,[size isize]);  
Figure,imshow(C),title('cropped');  
d=reshape(C,1,[]);  
disp('Reshapping done');  
fid = fopen('img.txt', 'wt');  
fprintf(fid, '%8d\n', d);  
disp('Text file write done');disp(' ');  
fclose(fid);
```

TESTER MODULE APPLYING VLSI TECHNIQUES

This module tests the flip-flop by generating the clock and D signal of the timing diagram above and dumping the Q and QN signals of the flip-flop. Its outputs are the flip-flop's inputs and viceversa.

```
//Tester module sends a periodic clock signal to the flip-flop  
module tester(q,qn,clk,d);  
input q,qn;  
output clk,d;  
reg clk,d;  
  
//Run the test once  
initial  
begin  
    clk=0;  
    //Dump results of the simulation to ff.vcd  
    $dumpfile("dff.vcd");  
    $dumpvars;  
    //Generate input signal d  
    d=0; #9 d=1; #1 d=0; #1 d=1; #2 d=0; #1 d=1; #12 d=0;  
    #1 d=1; #2 d=0; #1 d=1; #1 d=0; #1 d=1; #1 d=0; # 7 d=1;  
    #8 $finish;  
end  
  
//Generate periodic clock signal  
always  
begin  
    #4 clk=!clk;  
end  
endmodule
```

This module is behavioral too as we have initial and always blocks. You should be able to understand most of the code. However, there are a few new concepts here. The \$dumpfile and \$dumpvars commands tell the Verilog simulator (more on this ahead) to log the module's variables to the specified file, "dff.vcd" in this case. You may also be wondering what the #s do. These are Verilog delays. The delay the following instruction by a given amount of time. For example, #4 clk=!clk; within an always block changes "clk" every four time units from 0 to 1, producing a square wave. The time unit is a second by default. Without using delays there is no way of making the program work. This is the way to control time in the design. You may add delays to any instruction. For example, you could model the flip-flop's delay by adding some to its always block. It's now easy to understand how the d=0; #9 d=1; #1 d=0; #1 d=1; ... lines produce the D signal we want. Finally, the \$finish command tells the simulator to stop the simulation once the D signal was generated. If this command was omitted the simulation would continue indefinitely because this time the always block has no condition (there is no @ like in the flip-flop module).

TEST BENCH MODULE

This module just connects the tester module to the flip-flop module:

```
//Test bench connects the flip-flop to the tester module
module testbench;
    wire clk,d,q,qn;
    dff ff1(d,clk,q,qn);
    tester tst1(q,qn,clk,d);
endmodule
```

It is the most simple of the modules, but it's very different. This time it's structural code, that is, you define the structure of the circuit. It's like describing the circuit diagram. In this case the final circuit is simply the flip-flop connected to the tester. To create a flip flop use `dff ff1(d,clk,q,qn)`. First goes the module name, followed by the part name, which could be almost any string, followed by the wires that connect to the module in parenthesis. These must follow the order in the module's declaration. In a structural module we use wires. Regs are not necessary because they are defined inside the different modules.

COMPILING AND SIMULATING

Go ahead and copy/paste the modules into a text file, order doesn't matter. Call the file "dff.v". The .v extension is standard for Verilog files, but isn't required by the compiler. To compile open a Command Prompt at your working directory (where you saved the file). A quick way to open a command prompt at any directory is to hold shift and right-click the folder, then click "Open Command Window Here". Type:

```
iverilog -o dff dff.v
```

The "-o" tells Icarus Verilog to "save output to the following file". The output is then saved to "dff". This file is not executable. It has to be run using vvp, the Icarus Verilog simulator which is the one that actually produces simulation results, zeros and ones for each of the model variables, as a function of time. To run the simulation type:

```
vvp dff
```

This is what outputs the dff.vcd file with all the simulation results. However if you open this file with your text editor you'll see it's not easy to understand. To generate an easy-to-understand timing diagram from this file we use GTKWave.

GTKWave does have a GUI. To open it press Windows key and type "gtkwave". Then click File -> Open New Tab and chose the ffd.vcd file. Now you must add the variables in order to see their timing diagram. Click on "testbench" at the left (SST panel) and then select all the variables using Ctrl or Shift and "Insert" them.

If everything is okay you should get a timing diagram exactly as the one at the beginning of the tutorial, just like the following:

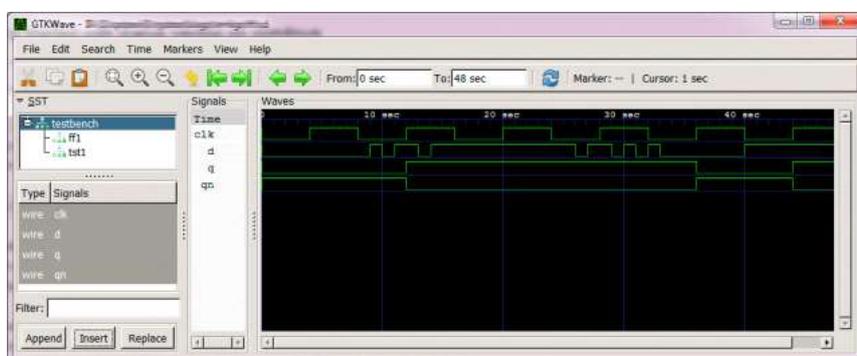


Fig. 6: Output Image of VLSI

When testing your programs you'll have to go to the compiling-simulating-loading process every few minutes. Remember you can use the up-down arrow keys while in the command prompt to access the last commands and compile/simulate. On GTKWave use File->Reload Waveform to reload the vcd file and refresh the timing diagrams without having to reload each variable. By using these tips the whole process will take you a few seconds. It's over. Now feel free to change the code around to see what happens. Mastering the use of delays, wires and rags takes some time. See Verilog_in_One_Day for a more in depth explanation of the language shown in [Fig. 6].

CONVERTING TEXT FILE INTO IMAGE

For converting text file into image, it needs to be converted from 1D matrix and then to image. The image size must same as previous for processing text into image. Then converted to 1D matrix using reshape command. Then write into text file. First, the text file is read using file read as vector (1D matrix). Then it has to be converted to matrix. Finally transpose to complete it as image.

```
fidh = fopen('img.txt');
Ah = fscanf(fidh, '%g %g', [1 inf]);
disp('Text file read done');
fclose(fidh);
S1h= vec2mat(Ah,ysize,ysize);
disp('Vector conversion done');
%c=imresize(S1,[128 128]);
Sh= transpose(S1h);
Jh=uint8(Sh);
disp('Image is ready');
imwrite(Jh,['newimage','.jpg']);
Figure,imshow(Jh),title('IMAGE form TEXT file');
```

OUTPUT

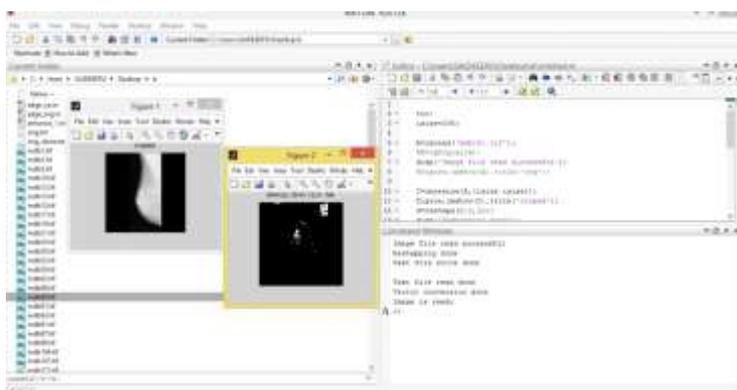


Fig. 7: Image Data base

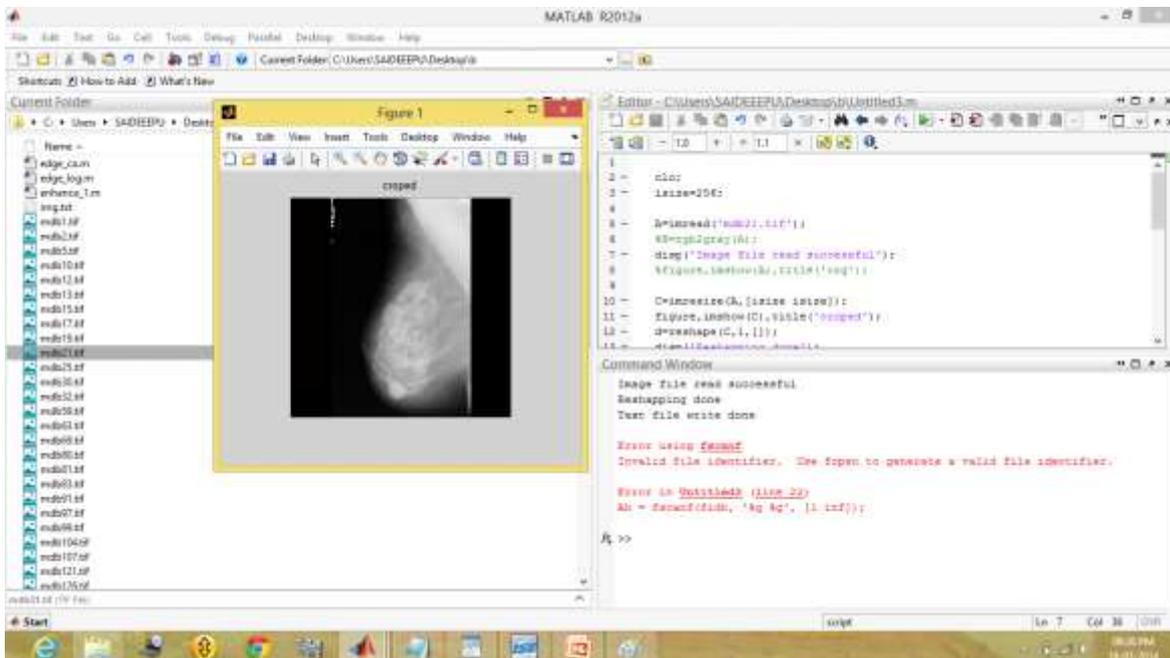
IMAGE DATABASE

The methods were tested with a total of 56 images (each of size 1024 1024 pixels at a resolution of 200 m) including 30 benign breast masses, 13 malignant, and 13 normal cases selected from the mini MIAS, database. The dataset includes circumscribed and spiculated cases in both benign and malignant categories. The cases in each category (benign and malignant) are further subdivided in the database as dense-glandular, fatty, and fatty-glandular types. Table I shows the numbers of various types of cases in each category. The mean values of the sizes of the masses are 1.07+_0.77 cm And 1.22+_0.85 cm for the benign and malignant categories, respectively. The radius of the smallest mass (malignant) is 0.34 cm, and that of the largest mass (benign) is 3.9 cm. The center of abnormality and an approximate radius of each mass are indicated in the database. The circular demarcation of masses as done in the database is not useful for confirming the results of mass detection as such a demarcation may also include normal fibroglandular tissues in the ROIs, particularly in spiculated cases. Hence, in the present work, the center of abnormality as indicated for each mass in the database was used to confirm the result of mass detection. Karssemeijer and to Brake, and Petrick etc. used a similar criterion for confirming the results of their mass detection algorithms. Karssemeijer and te Brake detected spiculated malignant tumors and lesions corresponding to architectural distortion using 19 malignant cases present in the MIAS database. Additionally, in the present work, the efficacy of the mass regions segmented in terms of benign versus malignant discrimination shown in [Fig. 7].

RESULTS

IMAGE TO TEXT CONVERSION

Initially the input is read on MATLAB. Then the image is cropped as show in the below shown in [Fig. 8].



Camel was secured in Fig. 8: Image to Text Conversion

IMAGE TO TEXT FILE

After reading the image, it is further converted into a text file and the intensity values of the text file is below .shown in [Fig. 9].



Fig. 9: Image To Text File

XILINX INPUT (TEXT)

The converted image is then given as input in the form of text on Xilinx and VERILOG is developed for the classification of cancer cells present in the image as shown in [Fig. 10] using Xilinx software.

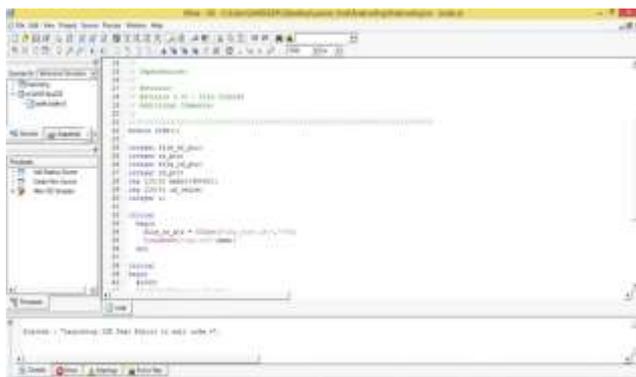


Fig. 10: XILINX Text

XILINX OUTPUT (WAVE FORMS)

The output for the classification of cancer cells in the text file is obtained in the form of wave forms on Xilinx as shown in [Fig.11]

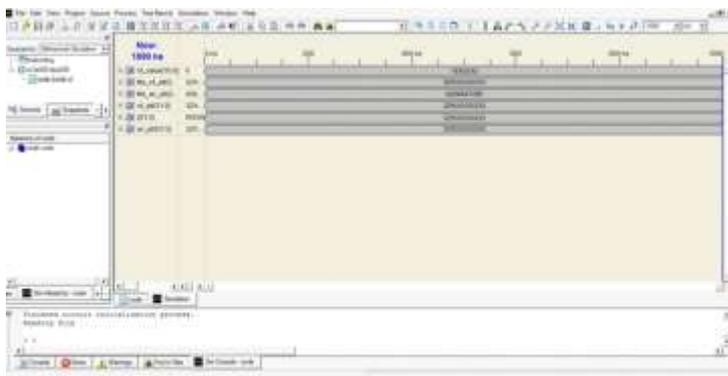


Fig. 11: XILINX Output

TEXT FILE OUTPUT INTO IMAGE

A text file is created according to the output waveforms on Xilinx. This text file is then read on MATLAB and converted into image in which cancer cells are classified as shown in [Fig. 12].

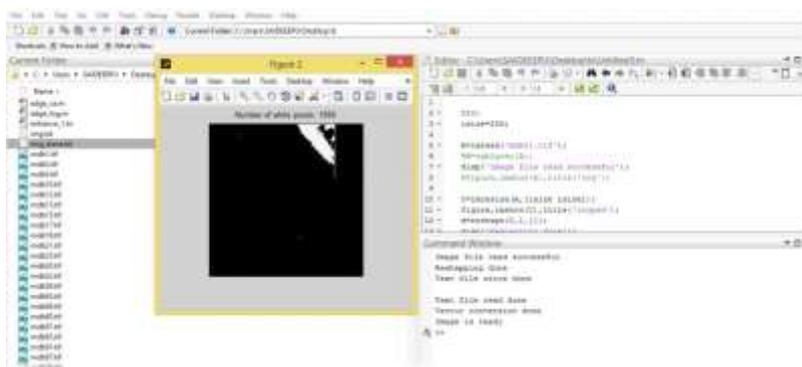


Fig. 12: Text File Output in to image

NUMBER OF CANCER CELLS AFFECTED

Finally after classification of cancer cells, the number of affected cells is calculated in MALAB and the result is shown in [Fig. 13].

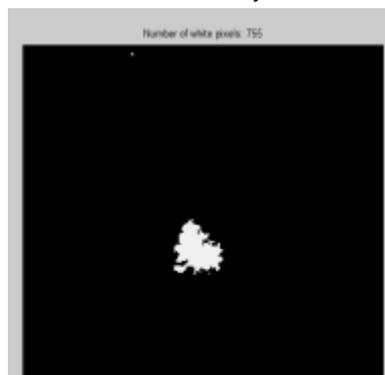


Fig.13: Number of cancer cells affected in the image.

CONCLUSION

Breast cancer is one of the major causes of death among women. Digital mammography screening programs can enable early detection and diagnose of the breast cancer which reduces the mortality and increases the chances of complete programs produce a great amount of mammographic images which have to be interpreted by radiologists. Due to the wide range of breast abnormalities' features some abnormalities may be missed or misinterpreted. There is also a number of false positive findings and therefore a lot of unnecessary biopsies. Computer-aided detection and diagnosis algorithms have been developed to help radiologists give an accurate diagnosis and to reduce the number of false positives. There are a lot of algorithms developed for detection of masses and calcifications. In this chapter, algorithms that are commonly used and the ones recently developed were presented. Over the years there has been an improvement in the detection algorithms but their performance is still not perfect. The area under the ROC curve is rarely above 90% which means that there are still many false positive outputs. Possible reason for such a performance may be the characteristics of breast abnormalities. Masses and calcifications are sometimes superimposed and hidden in the dense tissue which makes the segmentation of correct regions of interest difficult. Another issue is extracting and selecting appropriate features that will give the best classification results. Furthermore, the choice of a classifier has a great influence on the final result and classifying abnormalities as benign or malignant is a difficult task even for expert radiologists. Further developments in each algorithm step are required to improve the overall performance of computer-aided detection and diagnosis algorithms

CONFLICT OF INTEREST

There is no conflict of interest.

ACKNOWLEDGEMENTS

None

FINANCIAL DISCLOSURE

None.

REFERENCES

- [1] AntoBennet M, Sankar Babu G, Natarajan S, [2015] Reverse Room Techniques for Irreversible Data Hiding , Journal of Chemical and Pharmaceutical Sciences 08(03): 469-475.
- [2] AntoBennet, M , Sankaranarayanan S, Sankar Babu G, [2015] Performance & Analysis of Effective Iris Recognition System Using Independent Component Analysis, Journal of Chemical and Pharmaceutical Sciences 08(03): 571-576.
AntoBennet, M, Suresh R, Mohamed Sulaiman S, [2015] Performance & analysis of automated removal of head movement artifacts in EEG using brain computer interface, Journal of Chemical and Pharmaceutical Research 07(08): 291-299.
- [3] AntoBennet, M & JacobRaglend, [2012] Performance Analysis Of Filtering Schedule Using Deblocking Filter For The Reduction Of Block Artifacts From MPEQ Compressed Document Images, Journal of Computer Science, 8(09): 1447-1454.
- [4] AntoBennet, M & JacobRaglend, [2011] Performance Analysis of Block Artifact Reduction Scheme Using Pseudo Random Noise Mask Filtering, European Journal of Scientific Research, 66(1):120-129.
- [5] AntoBennet, M , Resmi R. Nair, Mahalakshmi V, Janakiraman G [2016] Performance and Analysis of Ground-Glass Pattern Detection in Lung Disease based on High-Resolution Computed Tomography, Indian Journal of Science and Technology, 09 (02):01-07.
- [6] AntoBennet, M & JacobRaglend, [2012] A Novel Method Of Reduction Of Blocking Artifact Using Machine Learning Metric approach, Journal of Applied Sciences Research, 8(5) :2429-2438.