IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 28, 2021, accepted February 5, 2021, date of publication February 11, 2021, date of current version March 3, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3058773

Preprocessing of Breast Cancer Images to Create Datasets for Deep-CNN

ABHIJITH REDDY BEERAVOLU[1], SAMI AZAM[1], MIRJAM JONKMAN[1], (Member, IEEE), BHARANIDHARAN SHANMUGAM[1], (Member, IEEE), KRISHNAN KANNOORPATTI[1], AND ADNAN ANWAR[2], (Member, IEEE)
[1]College of Engineering, IT, and Environment, Charles Darwin University, Casuarina, NT 0810, Australia
[2]Centre for Cyber Security Research and Innovation (CSIR), School of IT, Deakin University, Geelong, VIC 3216, Australia

Corresponding author: Sami Azam (sami.azam@cdu.edu.au)

ABSTRACT Breast cancer is the most diagnosed cancer in Australia with crude incidence rates increasing drastically from 62.8 at ages 35-39 to 271.4 at ages 50-54 (cases per 100,000 women). Various researchers have proposed methods and tools based on Machine Learning and Convolutional Neural Networks for assessing mammographic images, but these methods have produced detection and interpretation errors resulting in false-positive and false-negative cases when used in the real world. We believe that this problem can potentially be resolved by implementing effective image pre-processing techniques to create training data for Deep-CNN. Therefore, the main aim of this research is to propose effective image pre-processing methods to create datasets that can save computational time for the neural network and improve accuracy and classification rates. To do so, this research proposes methods for background removal, pectoral muscle removal, adding noise to the images, and image enhancements. Adding noise without affecting the quality of details in the images makes the input images for the neural network more representative, which may improve the performance of the neural network model when used in the real world. The proposed method for background removal is the ''Rolling Ball Algorithm'' and ''Huang's Fuzzy Thresholding'', which succeed in removing background from 100% of the images. For pectoral muscle removal ''Canny Edge Detection'' and ''Hough's Line Transform'' are used, which removed muscle from 99.06% of the images. ''Invert'', ''CTI_RAS'' and ''ISOCONTOUR'' lookup tables (LUTs) were used for image enhancements to outline the ROIs and regions within the ROIs.

INDEX TERMS Breast cancer, background removal, deep convolutional neural network (D-CNN), image enhancements, mammogram, mini-MIAS, pectoral muscle removal.

I. INTRODUCTION

Breast cancer is the most common type of cancer affecting women worldwide. The incidence and mortality rates vary among countries, based on factors such as environment, access to advanced medical care, income levels, etc., [1]. The mortality rates are increasing yearly in countries that have a larger 'low to middle-income' population, which could be explained by a lack of access to cost-effective resources [1]. Incidence rates are also increasing in several developed countries, such as Australia [2]. It is important to increase awareness about breast cancer and encourage women to participate in screening examinations, as early detection and diagnosis have the potential to save lives [3]. Mammography is considered the gold standard for regular screening and the Government of Australia provides free regular examinations (1 per 2 years) to women over 40 years [4]. After collecting the screening data, it is important to analyze these data and provide a diagnosis as accurately and quickly as possible. The analysis of the screening data requires skilled radiologists. Unfortunately, however, there is a shortage of radiologists in Australia and around the world [5], especially in regional areas and under-developed countries. This can lead to delays in diagnosis and treatment. Therefore, it is important to develop an intelligent system that can detect and diagnose abnormalities quickly and accurately.

Before developing an intelligent system, it is important to effectively pre-process mammographic images. This involves

The associate editor coordinating the review of this manuscript and approving it for publication was Jin-Liang Wang.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

removing the background, pectoral muscle, and the addition of noise along with the application of image enhancements. Several methods have previously been proposed by researchers for image segmentation (background removal and pectoral muscle removal) [6], but less research was done on methods for image enhancements in the pre-processing stage. The focus of this research is proposing effective methods for image segmentation and enhancements.

The processed images are analyzed to make sure the pixel quality and the region of interest (ROI) in the images are not adversely affected by the proposed methods. A histogram comparison is done between the original images and the processed images to check for deviations in quality and pixel values of the images. Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM) values are calculated to estimate the noise in the images and to evaluate whether this noise affects the quality of details in the image.

A. RESEARCH MOTIVATION

Many countries lack human resources and technology to provide timely services for patients in terms of detection, diagnosis, and treatment of breast cancer. For breast cancer time is a very important factor in saving lives. Many researchers have proposed methods and tools for detection and diagnosis, but these systems have often produced false positive and false negative cases. This research aims to improve the methods for diagnosis and detection to decrease instances of misdiagnosis. Developing a cost-effective and computationally fast system could help save lives in under-developed, developing, and even developed countries. To improve accuracy in extracting ROIs and regions within the ROIs for classification, a method is proposed for mammographic image segmentation and image enhancement to create input images for the D-CNN. These images will be used to create training, validation, and testing data for the D-CNN.

B. RESEARCH APPROACH

This research aims to propose effective image pre-processing methods that are computationally simple to implement. The proposed methods are used on mammographic images to remove as much unwanted area as possible and enhance the local details so that ROIs and regions within the ROIs can be detected easily.

After collecting mammographic images from various sources, MIAS digital mammogram database [7] is selected for research and development as shown in Fig. 1. Based on a thorough analysis of literature and a review of existing methods, this research proposes methods for image background and pectoral muscle removal, image enhancements, and image pre-processing for D-CNN of mammographic images. In the application of the proposed methods, the images are processed to remove artifacts and noise from the background. The background removed images are then processed to remove the pectoral muscle. During the pectoral muscle removal process, noise is reintroduced into these images so that the images can represent real-world scenarios. This will improve the performance of the neural network model when used in the real world. After removing all the unwanted parts from the images, image enhancements are applied so that ROIs and regions within the ROIs are highlighted during the pre-processing stage. After processing the images with the proposed methods, the input images for the D-CNN are ready to be used for training, validation, and testing. Results are collected separately for each step of the implementation of the proposed methods (i.e., background removal, pectoral muscle removal, and image enhancements). The collected results are used for the analysis of the proposed methods and comparison with existing methods. After obtaining the results and analysis, the time and cyclomatic complexity of the algorithms and programs that are used in this research is calculated, so that the computational time of the execution and quality of the algorithms and programs can be estimated and compared. Finally, the statistical significance of the obtained results is calculated using the T-distribution table and P-value, to support the conclusion.

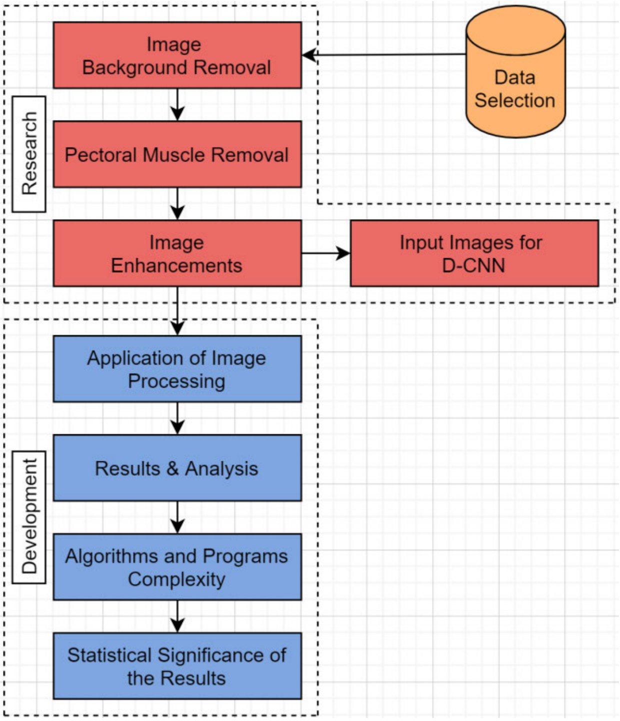


FIGURE 1. Research approach.

II. PREVIOUS WORK

A. BACKGROUND REMOVAL

Several methods have been proposed by previous researchers to remove the background from mammograms so that artifacts in the image do not interfere with the neural network. However, none of those methods try to remove the unwanted areas along with the artifacts. This research uses the "Rolling Ball Algorithm" [8] in combination with "Huang's Fuzzy Thresholding" and "Morphological Transformations" for background removal. The rolling ball algorithm has been applied in various areas of research but is not very often used in the field of mammogram analysis and breast cancer. In medical imaging, the rolling ball algorithm is mostly used for lung segmentation. Shaukat *et al.* [9] used the rolling ball algorithm to remove the background and irrelevant components for lung segmentation. They were able to achieve a sensitivity of 94.20% and 98.15% at the detection and classification stages, with only 2.19 false positives per scan. Their research indicated that accurate lung segmentation is important in enhancing the efficiency of lung nodule detection systems. El-Regaily *et al.* [10] used the rolling ball algorithm to reconstruct the lungs at the lung image segmentation stage by preserving the parts that are attached to the lung wall. Their convolutional neural network was able to achieve an accuracy of 89.895%, which is better than most of the other models. Tošić *et al.* [11] used the rolling ball algorithm for the detection of electromagnetic interference caused by LEDs in high-frequency radar range-doppler images. They used a rolling ball to remove uneven backgrounds from images, and they were able to identify and eliminate the background with a probability of 91%. An application of the rolling ball algorithm to mammograms was described by Basile *et al.* [12]. They used the rolling ball algorithm to highlight the primary regions inside the breast to detect micro-calcifications in mammograms. They achieved a true positive rate of 91.78%. This research uses the rolling ball algorithm to remove noise and identify intensity level artifacts in the mammographic image.

After processing the images with the rolling ball algorithm, this research uses "Huang's Fuzzy Thresholding Method" in combination with "Morphological Transformations" to remove artifacts, noise, and irrelevant portions from the mammographic images. Huang and Wang [13] proposed an image thresholding method to identify the fuzziness in an image. Their method used Shannon's Entropy Function [14] and Yager's Measure of Fuzziness [15] to quantify the fuzziness. Aja-Fernandez *et al.* [16] used a fuzzy thresholding method to overcome the limitations of segmentation for images that are corrupted with artifacts and noise. Sran *et al.* [17] proposed a framework that combines a saliency model with fuzzy thresholding to remove noise and artifacts so that tumor regions can be extracted accurately from brain MR images. They achieved a sensitivity of 97±3.0% in detecting the tumors. Our research uses threshold values determined by Huang's method, to create binary images so that artifacts and noise can be removed from the mammographic images during the background removal process.

After identifying the threshold values and creating the binary images, morphological transformations are applied to the images to ultimately remove the artifacts and the noise from the images. Lee and Wong [18] proposed an image segmentation method to remove noise from images based on mathematical grey-scale morphology. We have integrated some of the features of their method, such as erosion and dilation, with our proposed method. Hazarika and Mahanta [19] used morphological transformations (erosion and dilation) to identity the breast borders accurately and remove background from the images. They achieved an accuracy of 98.7%. Zebari *et al.* [20] also used these morphological transformations (erosion and dilation) to remove artifacts from their images, achieving an accuracy of 99.31%. The purpose of integrating the rolling ball algorithm, Huang's fuzzy thresholding, and morphological transformations in this research is to remove artifacts and noise from the mammographic images without affecting the quality and details of these images.

B. PECTORAL MUSCLE REMOVAL

Removing pectoral muscle is important to reduce the workload of the neural network and save computational time because it removes the parts from the image that are not required, thereby decreasing the size of the images. For muscle removal, detecting the edges is an integral part. To detect the edge, we use the "Canny Edge Detection" algorithm developed by John F. Canny in 1986 [21]. Rampun *et al.* [22] used canny edge detection to detect the initial contours and estimate the pectoral muscle boundary. Their pectoral muscle boundary estimation achieved a Jaccard Index [23] and a Dice Coefficient [24], [25] of 92.1 and 97.8, respectively. Taghanaki *et al.* [26] proposed a geometry-based muscle segmentation method to reduce the time and cost of computer-aided detection systems. They mentioned that removing muscle can also decrease false-positive rates because pectoral muscle and tumors in a mammographic image have the same density [26]. They have applied canny edge detection to detect and extract the breast contour and obtained Jaccard and Dice coefficients of 96±1.00% and 97.8±0.8%. They have achieved an overall accuracy of 95% in segmenting 322 MIAS image.

Biswas and Ghoshal *et al.* [27] used Sobel operators [28] in their research to detect blood cells in microscopic images. They used Sobel filters to increase the intensity of the edges, obtaining an accuracy of 93%. Kandhway *et al.* [29] found that Sobel operators are better at distinguishing the edge details in an image than the Laplacian and Canny edge detection operators. They compared the three operators by applying them to various medical images (mammograms, brain images, etc.). Our research uses two 3×3 Sobel Convolutional Kernels when applying canny edge detection.

To detect the edges generated from canny edge detection, this research implements the Hough Line Transform [30]. Bora *et al.* [31] used the Hough line transform to approximate the pectoral edge and segment the muscle from the mammogram using "texture gradient" and "Euclidean distance regression". Their method was able to remove pectoral muscle from 96.75% of their images. Shi *et al.* [32] also used the Hough line transform for detection of the muscle boundary and segmentation. Their method achieved an accuracy of 97.08%, which suggests that the Hough transform is good at detecting the muscle boundary.

C. ADDING NOISE

After removing the noise using the rolling ball algorithm during the background removal process, noise that should not affect the quality of the details in the images is added into the mammographic images during morphological transformation and pectoral muscle removal. Adding some noise to the images to create training data for the D-CNN will improve the performance of the neural network because the data will look more like real-world data. Injecting noise into the input of a neural network can also be considered as a form of data augmentation [33]. Training a neural network with the same type of data, if the images are normal, can cause overfitting issues, because the network can memorize all the training samples. Creating a training dataset with different levels of noise will make it less likely that the neural network memorizes the training data [33]. As Karimi *et al.* [34] noted in their research, deep learning models require more training data than traditional machine learning models. Deep neural networks often perform better during training and testing than when they are used in the real world because real-world data is not always clean. Bishop [35] found that the addition of noise to the input data during the training of the neural network can lead to improvements in the generalization performance. Neelakantan *et al.* [36] found that adding Gaussian noise to the gradient is effective when training deep networks. They found that the added noise can help optimize a neural network model that has many layers, in their case 20 layers. Zur *et al.* [37] also noted that training neural networks with noise reduces overfitting and improves the Area Under Curve (AUC) values, in this case by 0.02. Training the neural networks with noisy images will help the model to learn the de-noising process.

D. IMAGE ENHANCEMENT

Recently, several researchers (Kwok *et al.* [38], Ferrari *et al.* [39], Rampun *et al.* [22], Vikhe and Thool [40]) focusing on removing pectoral muscle from curved boundaries, found that this resulted in a reduction of accuracy rates. Soleimani and Michailovich [41] proposed a segmentation method that uses a convolutional neural network to detect the edges and the pectoral muscle boundary to segment it. They achieved a dice coefficient of 97.22±1.96%. The drawback however is that developing and using one CNN for pectoral muscle removal and then developing another CNN for detection and diagnosis can increase the computational costs and time when used in the real world. We believe that, rather than using a CNN for segmentation, computationally simpler methods for pectoral muscle segmentation at the pre-processing stage are more useful. If there are still some images with parts of muscle left after muscle removal using the Hough line transform, these images can be processed with the Deep-CNN developed for detection and diagnosis. This can create a more cost-effective system. Therefore, rather than focusing on the curved boundaries, we focus on removing as much muscle as possible using effective and simpler methods and tackle the remaining muscle portion by developing effective training mechanisms for neural networks to accurately detect the ROIs. To deal with the curved boundary issue, this research uses image enhancements by applying methods such as Look-up Tables (LUTs) so that the neural network can detect and extract the ROIs and regions within the ROIs.

Very few researchers have used image enhancement techniques at the pre-processing stage to create mammographic images that emphasize the details in the images for neural networks. LUTs are widely used for adjusting contrast or intensity characteristics between regions in an image. Sherrier and Johnson [42] used the concept of lookup tables for histogram equalization'', to equalize specific regions of chest images. Tellez *et al.* [43] used LUT based approach with a density histogram for characterizing the chromatic distributions to detect cell nuclei in tissue slides.

Lehmann *et al.* [44] noted that X-ray films are digitized with 12-bit quantization and then subsequently displayed by reducing them to 8-bit images, which results in a loss of important information. To avoid this, they transformed the images into 4096 displayable pseudo colors. They believe that pseudo coloring is essential for medical imaging. Their research implemented three image enhancement techniques which each have unique characteristics. Different LUTs are used to extract additional meaning from the images and highlight certain regions. The techniques are implemented using ``ImageJ'' [45] medical image processing software developed at the National Institutes of Health [46] and The Laboratory for Optical and Computational Instruments (LOCI, University of Wisconsin) [47]. Methods such as ``Invert LUT'', ``CTI_RAS LUT'' and ``ISOCONTOUR LUT'' are implemented using ImageJ, for image enhancements. Invert LUT inverts the gray level values in the image, from black to white and vice versa. CTI_RAS LUT draws a rainbow-themed boundary around the ROIs in the image. ISOCONTOUR LUT draws four contours of various colors (red, green, blue, yellow) based on the intensity levels in the image which helps to identify the regions within the ROIs. These methods use Look-Up Tables (LUTs) to determine the colors and intensity values to display in an image. John *et al.* [48] evaluated the use of a LUT for displaying chest CT images and concluded that the use of LUT methods has the potential to improve operational efficiency while achieving acceptable image quality. According to our knowledge, none of the researchers used LUTs for mammographic image enhancements to create training data for D-CNN.

E. SIGNIFICANCE OF IMAGE PRE-PROCESSING FOR D-CNN

Various types of public and private datasets were used by researchers as input for their CNNs for detection and diagnosis. Various image pre-processing methods were applied to the images of these datasets before feeding them into the CNNs. Tavakoli *et al.* [63] discussed the importance of image pre-processing to create training data. They mentioned that removing unwanted areas from the images will produce more accurate results. They have removed the artifacts, noise, pectoral muscle and enhanced the contrast of the images to improve the distribution of the pixel intensities. Their research aimed to classify the pixels of various regions of interest. They were able to achieve an accuracy of 94.68% and an AUC of 95%. They also performed experiments to evaluate the effects of pre-processing on the outcomes. Not using pre-processing methods produced an accuracy of 88.29% and AUC of 88%, while pre-processing produced 94.68% accuracy and an AUC of 95%. Ali *et al.* [65] also discussed the importance of pre-processing and calculated results with and without pre-processing. They were able to achieve an accuracy of 95.42% without preprocessing. With pre-processing an accuracy of 98.34% was obtained. This illustrates the usefulness of pre-processing. Our research uses various unique pre-processing methods to create the datasets for training.

Gao *et al.* [56] in their research implemented a 4-step image pre-processing procedure involving the identification of a bounding box that contained the tumor region. By enlarging and extracting the bounding box, normalizing the image intensity, and resizing the normalized image to 224*224, they obtained an accuracy of 90%. Unfortunately, these pre-processing procedures require considerable computational resources to extract the tumor region and require the coordinates of the tumor region before extraction. These methods cannot work when implemented in real-world scenarios where the location of the tumor is not available beforehand. We believe improving these methods through minimal preprocessing can improve accuracy.

Ribli *et al.* [57] proposed a Computer-Aided Detection (CAD) system based on a Faster R-CNN model for detecting and classifying lesions in a mammogram image. The system achieved an Area Under Curve (AUC) score of 0.85 and can detect 90% of the malignant lesions when used on a public dataset (INbreast). They did not perform any image pre-processing on the images. We believe that computational time for the CNN can be saved if pre-processing is performed also improving the detection and AUC score.

Saranyaraj *et al.* [58] proposed a D-CNN model to classify mammographic images. They noted that "image pre-processing is the most important step to obtain desired features and good classification rates". To decrease the computational cost, they have resized the images to 200*200 for training the D-CNN. The images in the datasets (DDSM) are normalized to help remove noise (maximum intensity value–minimum intensity value) before resizing and training the D-CNN. They have achieved a test accuracy of 96.23% and a classification accuracy of 97.46%. We believe the results can be made more robust by adding some noise into the images for training to represent real-world scenarios so that the D-CNN can produce good results when used in the real-world [33], [34].

Arevalo *et al.* [60] in their research found that pre-processing is important to enhance the characteristics of the mammographic images so that ROI extraction from the images can be improved. They noted that data augmentation by creating datasets through the application of various transformations can prevent overfitting issues. They also noted that performing global and local contrast normalization can both improve performance and reduce training time. Their research achieved an AUC of 86%. We believe this performance can be improved through the implementation of better image processing methods making it easier to extract the ROIs and regions within the ROIs.

Dubrovina *et al.* [61] used color coding in their research to highlight the details (pectoral muscle, fibro glandular tissue, breast tissue, background, etc.,) in the mammographic images so that sufficient local information can be captured to classify the pixels belonging to different tissues and regions. They created datasets using color-coded images to train the neural network to classify different tissues and regions in the images. They were able to achieve faster computation while maintaining the same classification accuracy. This research aims to create datasets for training, validation, and testing the D-CNN to improve the performance and increase the accuracy when using the system in real-world scenarios. This is done by using "rolling ball algorithm", "Huang's fuzzy thresholding", "Canny edge detection", "Hough line transform", "adding noise to the images", "applications of Look Up Tables (LUTs) for enhancements" making it easier to detect the ROIs and regions within the ROIs.

III. PROPOSED METHOD

A. DATA COLLECTION

Mammogram images from the "Mammographic Image Analysis Society" (MIAS) Digital Mammogram Database [7] are used for implementing the proposed methods. The dataset consists of 322 (1024 * 1024 Pixels) images. The dataset also provides various important details about the images, such as:

a. Reference Numbers for the images in the MIAS database.

b. Coordinates (x, y) for the center of abnormality.

B. BACKGROUND REMOVAL

Fig. 2 shows the entire background removal process and the methods involved.

1) ROLLING BALL ALGORITHM

The rolling ball algorithm is based on a concept described by Stanley Sternberg in his article "Biomedical Image Processing" [7]. The algorithm is applied through a python program that is ported from ImageJ's [45] "Background Subtractor" software tool.

As illustrated in Fig. 3, the algorithm works by using a ball of a given radius and rolling it over the surface of the image. It identifies a smooth continuous background in the mammographic image. The radius of the ball should be at least as large as the radius of the largest object, based on intensity levels in the image. The algorithm is implemented through four steps (see Fig. 3).

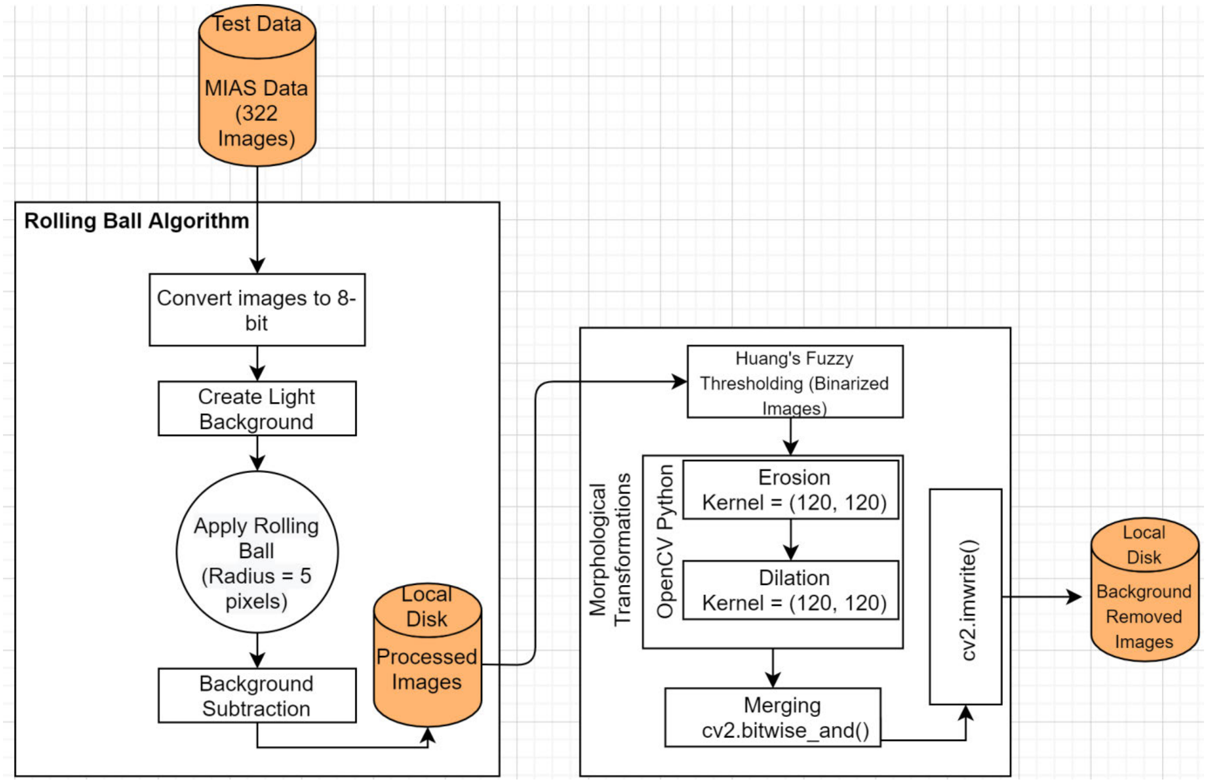

Test Data
MIAS Data (322 Images)
Rolling Ball Algorithm
Convert images to 8-bit
Create Light Background
Apply Rolling Ball (Radius = 5 pixels)
Background Subtraction
Local Disk
Processed Images
Huang's Fuzzy Thresholding (Binarized Images)
Morphological Transformations
OpenCV Python
Erosion Kernel = (120, 120)
Dilation Kernel = (120, 120)
Merging cv2.bitwise_and()
cv2.imwrite()
Local Disk
Background Removed Images

FIGURE 2. Background removal process.

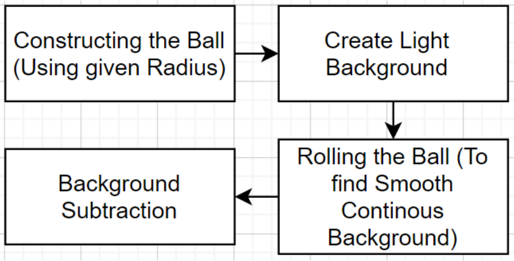

Constructing the Ball (Using given Radius)
Create Light Background
Rolling the Ball (To find Smooth Continous Background)
Background Subtraction

FIGURE 3. The workflow of rolling ball algorithm.

a: CONSTRUCTING THE BALL

As shown in Fig. 3, first a ball is constructed with a given radius. For our purpose, because of the size of the artifacts, a radius of 5 pixels is used for the ball as can be seen from Fig. 4.

Algorithm 1 provides the details of the construction of the ball. The algorithm makes use of variables such as "ball_radius", "arc_trim_per", "shrink_factor", "rsquare", and "xtrim". The values in these variables are used to construct the ball. The variable 'shrink_factor' is used to shrink the rolling ball by a certain factor before rolling the ball. The variable 'arc_trim_per' (i.e., trimming the arc) is used to trim off some percentage from each side of the rolling ball to create **patches** on the ball. When the ball is rolled on the image, these patches are used to identify the intensity values of the artifacts. A shrink_factor of 1 pixel and an arc_trim_per of 24% is used if the radius of the ball is less than 10 (in this case it is 5 pixels). Using the shrink_factor and ball_radius values, the ball is constructed. After constructing the ball, the values 'rsquare' and 'xtrim' are used to determine the radius of the shrunken arc and the number of points to be removed from each arc. These values are used to create **patches** on the surface of the ball.

Algorithm 1 Construct the Ball

```
BEGIN
    FUNCTION build(ball_radius, arc_trim_per):
        small_ball_radius < - ball_radius / shrink_factor
        IF small_ball_radius < 1 THEN
            small_ball_radius < - 1
        ENDIF
        rsquare < - small_ball_radius * small_ball_radius
        xtrim < - int(arc_trim_per * small_ball_radius)
            / 100
END
```

b: CREATE LIGHT BACKGROUND

As shown in Fig. 3, after constructing the ball, a light background is applied to the mammographic images. This highlights the dark artifacts and their locations, as observed in Fig. 4. Rolling the ball after applying a light background will make it easy for the ball to calculate the pixel values that need to be subtracted from the image to remove the artifacts.

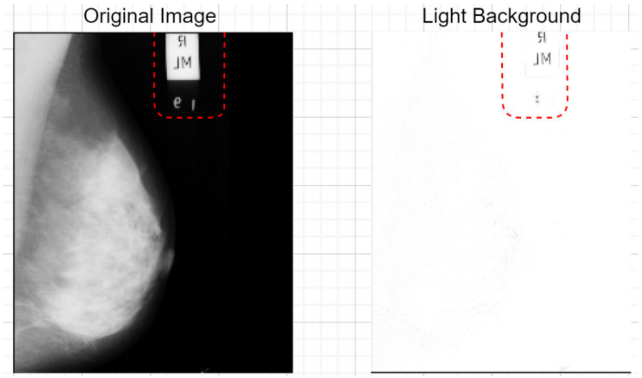


FIGURE 4. Converting 'Original image' to 'Image with light background'.

The light background is applied using the OpenCV (Open Computer Vision) python program.

c: ROLLING THE BALL

After constructing the ball and creating the light background on the original image, the ball is rolled over the light background image to identify the smooth continuous background and the dark artifacts, as shown in Fig. 3, Using the **patches** on the surface of the ball, the pixel values at the location of the artifacts are obtained. The "Rolling Ball Algorithm" is applied during this step. As highlighted in Fig. 4 under "Light Background", the pixel values at the highlighted location are obtained by the ball.

To identify the exact pixel values (0,255) and their coordinates (x, y), an additional dimension 'z' (height), based on the gray intensity values (0 - 255) at the pixel locations is plotted along the (x, y) dimensions. The ball is rolled so that a **patch** on the ball can be tangent to one or more points in the image. If any point in the image is on or below the patch, it is part of the background. Here, 0 indicates black, and 255 indicates white.

Algorithm 2 provides the details of how the ball is rolled over the surface of the image. The variables next_line_to_read and next_line_to_write_in_cache are used by the ball to read each pixel in the image, and if the intensity value of the pixel (i.e., height) is greater, then the pixel location is read and stored in the variable 'src'. After identifying the location, it is stored in 'cache' and the process is repeated till all the pixel values are identified and stored in the cache. Algorithm 2 is applied using a python program that uses the 'NumPy' library to obtain the image 'array'.

After rolling the ball, smooth continuous background and the associated pixel values are identified using Algorithm 3. 'xp' and 'yp' represent the pixel intensity values identified at coordinates (x, y). x_0 and y_0 represent the zero value and x_end +1 and y_end +1 represent 255+1 (256). For each point in the cache that is identified, the intensity values (z) for the smooth continuous background are identified.

Algorithm 2 Rolling the Ball

```
BEGIN
    FUNCTION roll_ball(ball, array):
        for y in range(-radius, height + radius) DO
            next_line_to_write_in_cache <- (y + radius) % ball_width
            next_line_to_read <- y + radius
            IF next_line_to_read < height THEN
                src <- next_line_to_read * width
                dest <- next_line_to_write_in_cache * width
                cache[dest:dest + width] <- pixels[src:src + width]
                p <- next_line_to_read * width
                for x in range(width) DO
                    pixels[p] <- -float('inf')
                    p + = 1
                ENDFOR
            ENDIF
        ENDFOR
END
```

Algorithm 3 Finding Smooth Continuous Background

```
BEGIN
    for yp in range(y_0, y_end + 1) DO
        cache_pointer <- (yp % ball_width) * width + x_0
        bp <- x_ball_0 + y_ball * ball_width
        for xp in range(x_0, x_end + 1) DO
            z_reduced <- cache[cache_pointer] - z_ball[bp]
            IF z > z_reduced THEN
                z <- z_reduced
            ENDIF
            cache_pointer + = 1
            bp + = 1
        ENDFOR
        y_ball + = 1
    ENDFOR
END
```

d: SUBTRACT BACKGROUND

As shown in Fig. 3, the mammographic images with bright background and dark artifacts are converted into an image with a light background. Then a ball of radius 5 pixels is rolled to identify the smooth continuous background and the location of its pixels. The python program will then subtract the identified pixels from the original image. This will subtract the dark artifacts from the image, as can be seen in Fig. 5 under "Background Subtraction".

'background_pixels' in Algorithm 4 describes the pixel values identified from Algorithm 3. The identified values are subtracted from each pixel value in the image. The value is assigned as 0 (i.e., white) if the pixel subtraction result is less than 0 and as black if the subtraction result is greater than 255 (i.e. black).

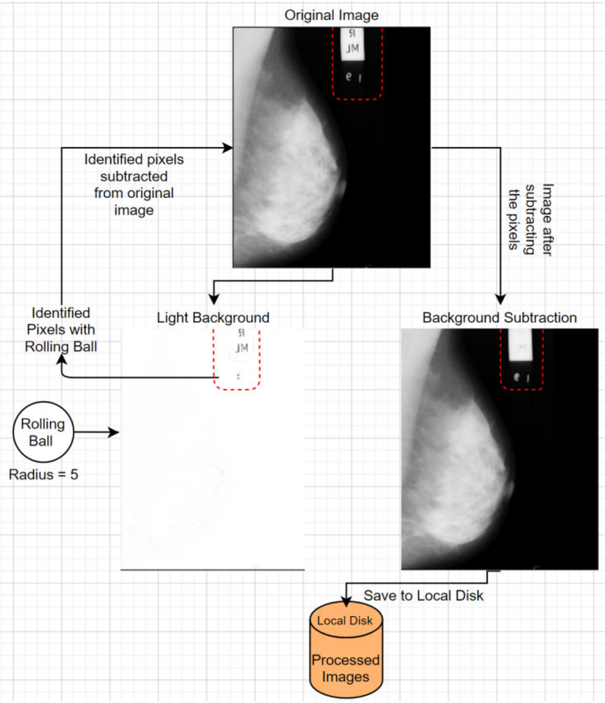


FIGURE 5. Background subtraction.

Algorithm 4 Background Subtraction

```
BEGIN
    for p in range(len(pixels)) DO
        value <- (pixels[p] & 0xff)
            - (background_pixels[p] + 255)
        IF value < 0 THEN
            value <- 0
        ENDIF
        IF value > 255 THEN
            value <- 255
        ENDIF
        pixels[p] <- numpy.int8(value)
    ENDFOR
    RETURN numpy.reshape(pixels, array.shape)
END
```

2) HUANG'S FUZZY THRESHOLDING

The processed images that are stored on the local disk after applying the rolling ball algorithm and background subtraction, are converted to binary images using the threshold values generated using Huang's method [13], see Fig. 6. The method is applied using a python program.

This image thresholding method uses the concept of "fuzzy sets" and the "definition of membership" function to measure the fuzziness in an image and obtain the appropriate threshold value. Huang's method uses "Shannon's Entropy Function" and "Yager's Fuzzy Measure" to measure the fuzziness in the image. The method aims to minimize the measure of fuzziness in any input image. The methods and steps proposed by Huang and Wang [13] are used in this research.

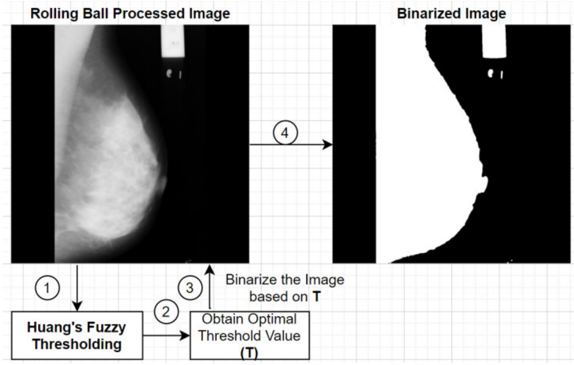

Rolling Ball Processed Image
Binarized Image
4
1
3
Binarize the Image based on T
2
Huang's Fuzzy Thresholding
Obtain Optimal Threshold Value (T)

FIGURE 6. Image binarization.

a: MEMBERSHIP FUNCTION

Let **A** denote a fuzzy set, a membership function for **A** on the universe of discourse **D** is defined as $\mathbf{u_A : D \rightarrow [0, 1]}$, where each element of **D** is mapped to a value between 0 and 1.

b: IMAGE THRESHOLDING

Let **X** denote an image set of size **M x N** with **L** levels and $\mathbf{x_{mn}}$ be the gray level of an **(m, n)** pixel in **X**. Let $\mathbf{u_X(x_{mn})}$ be the membership function.

The image set **X** in the notation of fuzzy set can be defined as,

$$\mathbf{X} = \{(\mathbf{x_{mn}}, \mathbf{uX(x_{mn})})\} \tag{1}$$

where $\mathbf{u_X(x_{mn})}$ is in the interval **[0,1],** which represents the fuzziness of the **(m, n)** pixel in **X.**

$\mathbf{m = 0, 1, 2, \ldots, M\text{-}1}$ and $\mathbf{n = 0, 1, 2, \ldots, N\text{-}1}$

The membership function is used to define the relation between the pixels in **X** and its corresponding region (i.e., background or object).

Let $\mathbf{u_0}$ and $\mathbf{u_1}$ be the average grey level values of the background and the object. Let **g** denote the gray level in an input image and **h(g)** the number of occurrences of this the gray level.

For a given threshold **t,** the target values of $\mathbf{u_0}$ and $\mathbf{u_1}$ (i.e., background and object) can be defined as **(2) and (3)**

$$u_0 = \frac{\sum_{g=0}^{t} gh\,(g)}{\sum_{g=0}^{t} h(g)} \tag{2}$$

$$u_1 = \frac{\sum_{g=t+1}^{L-1} gh\,(g)}{\sum_{g=t+1}^{L-1} h\,(g)} \tag{3}$$

The membership function depends on the difference between the gray value at a pixel in **X** and the target values of the background and object, defined as:

$$\begin{aligned} u_x\,(x_{mn}) &= \frac{1}{1+\frac{|x_{mn}-u_0|}{c}} \qquad if\ x_{mn} \le t \\ &= \frac{1}{1+\frac{|x_{mn}-u_1|}{c}} \qquad if\ x_{mn} > 1 \end{aligned} \tag{4}$$

C = constant value, such that $\frac{1}{2} \le u_x\,(x_{mn}) \le 1.$

The pixel **(m,n)** should be either part of the background or part of the object for a given threshold **t**.

Using **eq (4)** and Shannon's Function [14] or Yager's Measure [15], the fuzziness in the input image can be measured. The minimum measure is used to determine the optimal threshold value. We have used Shannon's function for our research.

c: SHANNON'S ENTROPY FUNCTION

Entropy **(E)** is used as a measure of fuzziness. It is defined using Shannon's Function **(S)** as,

$$E(X) = \frac{1}{MNln2} \sum_{g} S(u_x(g))\, h(g) \tag{5}$$

where g = 0, 1, …., L-1.

E(X) = **0**, if $u_x(x_{mn}) = 0$ **or 1** for all **(m, n)**

E(X) = **1,** if $u_x(x_{mn}) =$ **0.5** for all **(m, n)**

This research has used **eq (4)** and **eq (5)** to obtain the optimal threshold values of the images. The equations were implemented through a python program, represented by Algorithm 5.

Algorithm 5 Huang's Fuzzy Threshold

```
BEGIN
    threshold <- -1
    min_ent < − float(“inf”)    //minimum entropy
    for it in range(254) DO
        ent <- 0.0    //entropy
        for ih in range(it) DO
            mu_x <- 1.0 / (1.0 + term * math.fabs
            ih - mu_0[it]))    //eq(4)
            IF (not ((mu_x < 1e-06)
            OR (mu_x > 0.999999))) THEN    //eq(5)
                ent <- ent+ data[ih] * (-mu_x * math. log
                (mu_x) - (1.0 - mu_x) * math.
                log(1.0 - mu_x))
            ENDIF
        ENDFOR
        IF (ent < min_ent) THEN
            min_ent <- ent
            threshold <- it
        ENDIF
    ENDFOR
    RETURN threshold
END
```

The obtained threshold value from Algorithm 5 is used to binarize the mammographic images processed with the rolling ball algorithm, as shown in Fig. 6.

3) MORPHOLOGICAL TRANSFORMATIONS

Finally, morphological transformations are applied to the binarized images (Fig. 6), to remove the artifacts. The transformations are applied using the OpenCV python program. This research has used "erosion" followed by "dilation" as transformation operations.

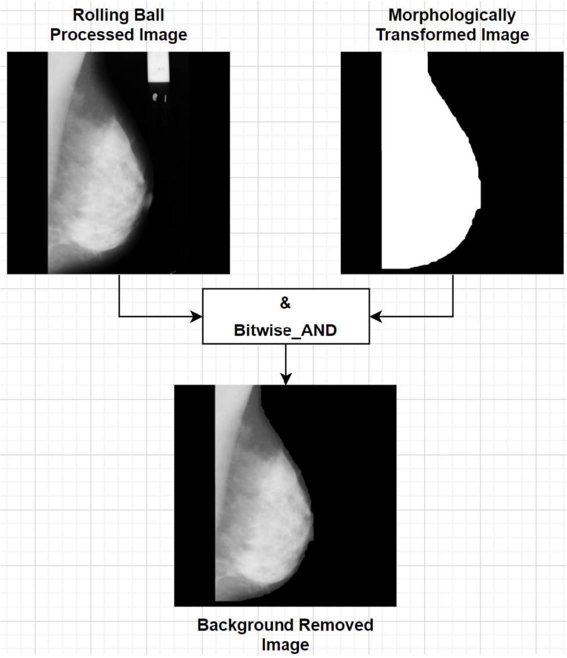

Rolling Ball Processed Image
Morphologically Transformed Image
&
Bitwise_AND
Background Removed Image

FIGURE 7. **Merging.**

a: EROSION

The erosion operation shrinks the mammographic image in such a way that the bright areas get smaller and the dark areas get bigger. The operation (**eq. 6)** is performed using a kernel of size (120,120). This kernel, which has an 'anchor point' (i.e., the center of the kernel), is used to convolve (scan over) the mammographic image. After convolving, the pixels in the image are replaced with the 'minimal pixel value' that is computed using the kernel.

Let **X** be the set of pixels in a binary image and **B** a kernel of size **n** which is scanned over the image to compute the minimal pixel value overlapped by the kernel and replace the pixels in the image that are below anchor point **z** with that minimal pixel value.

$$X \ominus B = \{z : B_z \subseteq X\} \tag{6}$$

where $B_z = \{b + z : b \epsilon B\}$ is the translation of kernel **(B)** by its anchor point **z.**

b: DILATION

The dilation operation is performed on the eroded image. The operation expands the bright areas in the image. A kernel **B** of size (120,120) is scanned over the image to compute the maximal pixel value overlapped by the kernel and replace the image pixel in the anchor point **z** with that maximal value **(eq. 7).**

$$X \oplus B = \{x + b : x \epsilon X, b \epsilon B\} \tag{7}$$

4) MERGING

As observed in Fig. 7, the rolling ball processed image and the morphologically transformed image are merged using the **bitwise AND** operator to remove the artifact from the image.

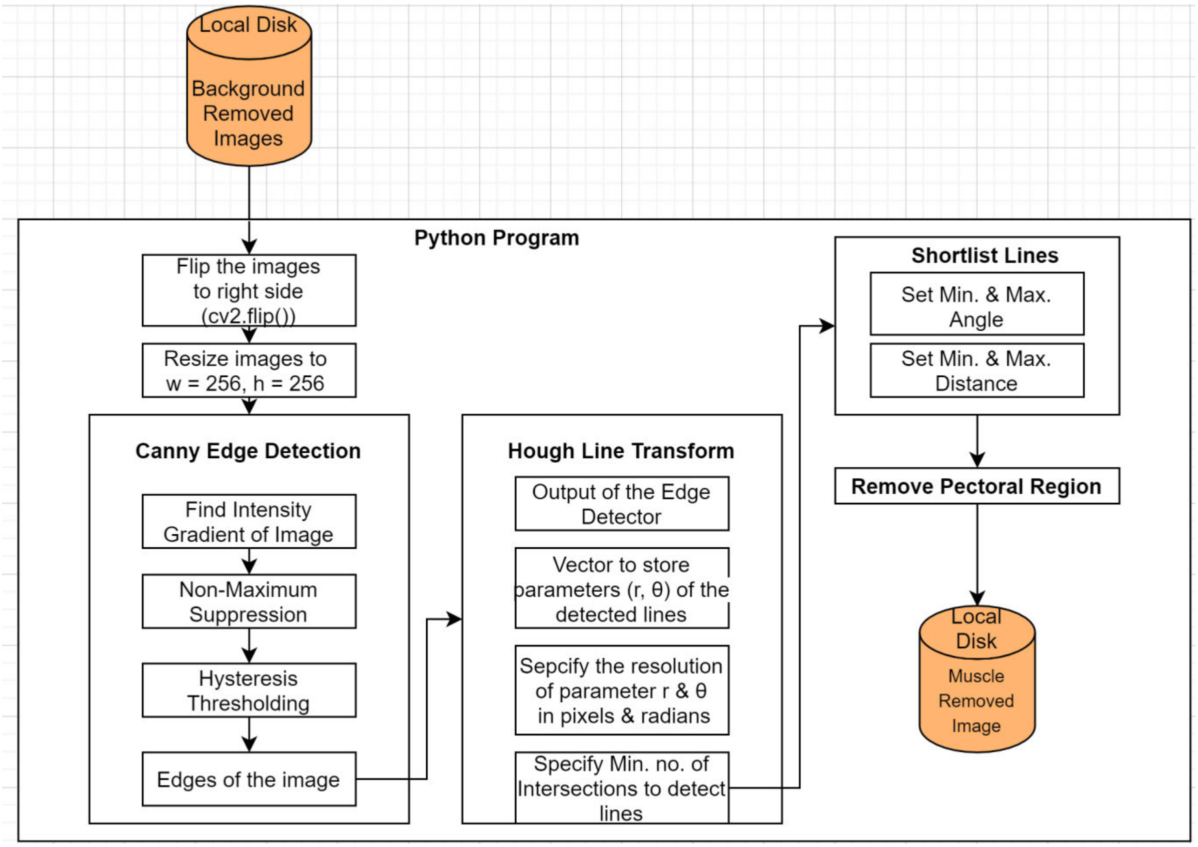

Local Disk
Background Removed Images
Python Program
Flip the images to right side (cv2.flip())
Resize images to w = 256, h = 256
Canny Edge Detection
Find Intensity Gradient of Image
Non-Maximum Suppression
Hysteresis Thresholding
Edges of the image
Hough Line Transform
Output of the Edge Detector
Vector to store parameters (r, θ) of the detected lines
Sepcify the resolution of parameter r & θ in pixels & radians
Specify Min. no. of Intersections to detect lines
Shortlist Lines
Set Min. & Max. Angle
Set Min. & Max. Distance
Remove Pectoral Region
Local Disk
Muscle Removed Image

FIGURE 8. Pectoral muscle removal process.

After merging both images, the background is removed, and the images are stored on a local disk.

C. PECTORAL MUSCLE REMOVAL

Every area in the mammographic image that is not required should be removed to limit memory requirements and computational time for the neural network. After removing the artifacts, the only part which is not required is the muscle. Rather than extending the neural network or increasing the processing power of the neural network, it is important to develop computationally simple methods to remove as much muscle as possible in the image pre-processing stage. As shown in Fig. 8, this research has used ''Canny Edge Detection'' and ''Hough Line Transform'' to remove the muscle.

Firstly, the images generated by the merging operation (in Fig. 7) will be flipped to the right side and then resized, because it will be faster and easier to automate the process of removing the muscle if all the mammographic images are pointing in the same direction and are smaller in size. This makes it easier to detect the muscle boundary.

1) RIGHT-SIDE FLIPPING & RESIZING

The images are flipped and resized using the OpenCV (Open Computer Vision) python program. The images are resized from 1024 × 1024 pixels into 256 × 256 pixels.

2) CANNY EDGE DETECTION

The Canny edge algorithm [21] is used to detect the edges in the mammographic image. The muscle boundary is obtained using this algorithm. The algorithm is implemented using a python program.

Step 1 (Calculating the Gradient):

The intensity gradient of the image is calculated to detect the intensity of the edge and its direction. Edges occur when the intensity value of the pixel changes in the images. To detect these changes, **Sobel Kernels (S)** are applied along the horizontal **(x)** and vertical directions **(y)**. **Two** (3 × 3) Sobel kernels **(K_x)** and **(K_y)** are used to detect the edges and compute the intensity. The kernels are convolved with the image to find the horizontal **(S_x)** and vertical **(S_y)** change in intensity. As shown in Fig. 9, two images are generated after the convolution-based on **(S_x)** and **(S_y)**. These images are used to compute the edge intensity and direction.

$$k_x = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \quad k_y = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

$$\boldsymbol{IntensitGradeint} = \sqrt{S_x^2 + S_y^2} \tag{8}$$

$$\boldsymbol{Direction} = \boldsymbol{arctan}(\frac{S_y}{S_x}) \tag{9}$$

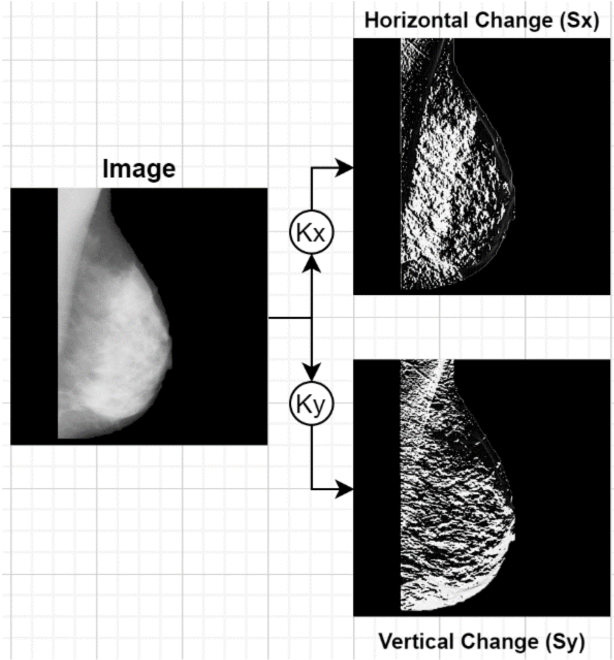


FIGURE 9. Horizontal & vertical change.

Algorithm 6 Calculating Edge Gradient and Direction

```
BEGIN
    FUNCTION sobel_filters(img):
        Kx <- np.array([[−1, 0, 1], [−2, 0, 2],
        [−1, 0, 1]], np.float32)
        Ky < − np.array([[1, 2, 1], [0, 0, 0],
        [−1, −2, −1]], np.float32)
        Sx <- ndimage.filters.convolve(img, Kx)
        Sy <- ndimage.filters.convolve(img, Ky)
        G <- np.hypot(Sx, Sy)
        G <- G / G.max() * 255
        D <- np.arctan2(Sy, Sx)
        RETURN (G, D)
END
```

The above equations are implemented through a python program using the "SciPy" and "NumPy" libraries. Algorithm 6 represents equations **(8)** and **(9)** along with the kernels $\mathbf{K_x}$ and $\mathbf{K_y}$.

where **G** is Intensity Gradient and **D** is Direction.

Step 2 (Non-Maximum Suppression):

After computing the edge intensity and direction obtaining an image with all the edges, any pixels that are not required (i.e., not an edge) should be removed. As demonstrated in Fig. 10, non-maxima suppression is used to remove these pixels and extract the required edges. This step looks at those pixels with maximum value that are pointed towards the direction of the edge.

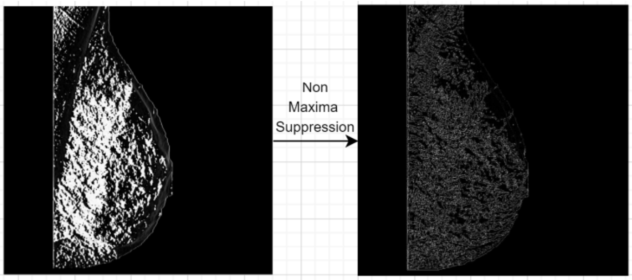


FIGURE 10. Non-maxima suppression.

Algorithm 7 Non-Maximum Suppression

```
BEGIN
    IF (0 <= angle[i,j] < 22.5) OR (157.5
      <= angle[i,j] <= 180) THEN
          q <- img[i, j+1]
          r <- img[i, j−1]
    ELSEIF (22.5 <= angle[i,j] < 67.5) THEN
          q <- img[i+1, j−1]
          r <- img[i−1, j+1]
    ELSEIF (67.5 <= angle[i,j] < 112.5) THEN
          q <- img[i+1, j]
          r <- img[i−1, j]
    ELSEIF (112.5 <= angle[i,j] < 157.5) THEN
          q <- img[i−1, j−1]
          r <- img[i+1, j+1]
    ENDIF
    IF (img[i,j] >= q) AND (img[i,j] >= r) DO
          Z[i,j] <- img[i,j]
    ELSE THEN
          Z[i,j] < − 0
    ENDIF
    RETURN Z
END
```

Using Algorithm 7, non-maximum suppression is performed. Here, **(i, j)** represents the pixel that is being processed, whereas **(i, j−1)**, **(i, j+1)**,**(i+1, j)**, **(i−1, j)**, etc., represent the pixels surrounding **(i, j)**. The values of **q, r** are set as 255 (i.e. white) to identify the pixels that have more intensity. As every pixel is processed in all directions (using angles), only the pixels with high intensity are kept, see Fig. 10.

Step 3 (Hysteresis Thresholding):

This step is used to distinguish between the edges and identify the boundary of the breast and the muscle, as shown in Fig. 11. It requires two threshold values **minVal** and **maxVal** to extract the intensity gradient values.

Edges with the intensity gradient > **maxVal,** are subsequently considered as edges. If the intensity gradient of the edges < **minVal,** they will be considered non-edges. Based on these values, pixels are categorized into **strong, weak, other.** The **weak** pixels are converted to **strong** pixels based on the conditions provided in the Algorithm. 8, where (**M, N**) represents the size of the image and (**i, j**) represents the value of the location in the image. This algorithm produces an image with strong edges as the output.

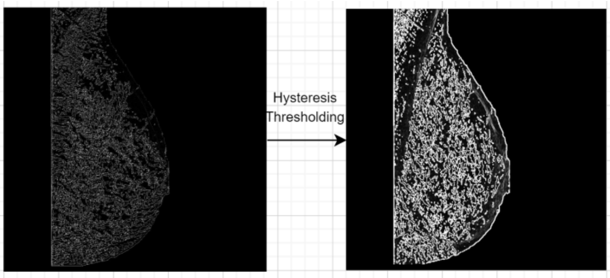


FIGURE 11. **Hysteresis thresholding.**

Algorithm 8 Hysteresis Thresholding (minVal & maxVal)

```
BEGIN
  FOR i in range(1, M-1) DO
    FOR j in range(1, N-1) DO
      IF (img[i,j] = minVal) THEN
        TRY
          IF ((img[i+1, j−1] = maxVal) OR
             (img[i+1, j] = maxVal) OR (img[i+1, j+1]
             = strong)
             OR (img[i, j−1] = maxVal) OR (img[i,
             j+1] = strong)
             OR (img[i−1, j−1] = strong) OR (img[i−1, j]
             = strong)
             OR (img[i−1, j+1] = strong)) THEN
            img[i, j] <- strong
          ELSE THEN
            img[i, j] <- 0
          ENDIF
      ENDIF
    ENDFOR
  ENDFOR
  RETURN img
END
```

3) HOUGH LINE TRANSFORM

This research focuses on developing methods that are computationally simple and easy to implement. These methods are used to remove the unwanted areas from the mammographic images, as much as possible. To detect the edges generated from canny edge detection, this research has implemented the "Hough Line Transform" [30].

The Hough line transform method identifies the lines in the images. Various lines are constructed in the image. These lines are used to identify the muscle boundary. The lines are expressed using the "polar coordinate system" (Fig. 12) and the lines are constructed using equation 10, where **r,**$\boldsymbol{\theta}$ represents distance and angle. Hough line transform is applied using a python program.

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r}{\sin\theta}\right) \tag{10}$$

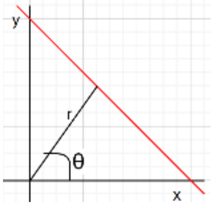


FIGURE 12. **Polar coordinate system.**

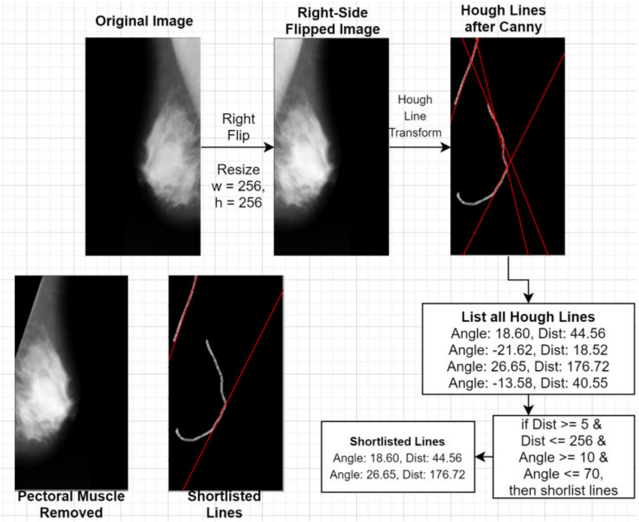

Original Image
Right-Side Flipped Image
Hough Lines after Canny
Right Flip
Resize w = 256, h = 256
Hough Line Transform
List all Hough Lines
Angle: 18.60, Dist: 44.56
Angle: -21.62, Dist: 18.52
Angle: 26.65, Dist: 176.72
Angle: -13.58, Dist: 40.55
if Dist >= 5 & Dist <= 256 & Angle >= 10 & Angle <= 70, then shorlist lines
Shortlisted Lines
Angle: 18.60, Dist: 44.56
Angle: 26.65, Dist: 176.72
Pectoral Muscle Removed
Shortlisted Lines

FIGURE 13. **Pectoral muscle removal process.**

Hough line transform looks for intersections between the points in the image to draw the lines. It identifies whether the intersections are above a certain threshold, as demonstrated in Fig. 16. After extracting the edges in the image using the canny edge method, Hough transform is used to compute the **distance** and **angle** to the muscle boundary from the **origin**(the origin is the center of the image), see Fig. 13 ("List all Hough Lines"). The lines are constructed using the distance and angle parameters. The line representing the muscle boundary is **shortlisted** based on the conditions that are set.

The conditions are set based on the size of the image and the position of the muscle boundary (i.e., **Width** $=$ **256, Height** $=$ **256**). All the parameters are measured from the origin. These parameters can be adjusted based on the size of the images and the size of the muscle boundary.

Conditions to shortlist the line (i.e., muscle boundary), are for most of the images:

Minimum angle = 10 **;** Maximum angle = 70

Minimum distance = 5; Maximum distance = 256

Minimum and Maximum values are set based on the size of the image (i.e. 256 × 256) and the position of the muscle boundary from the center of the image (i.e. the angle). The position is mostly below 90^0 from the center.

If **distance** $>=$ **minimum distance** & **distance** $<=$ **maximum distance** & **angle** $>=$ **minimum angle** & **angle** $<=$ **maximum angle.**

A line that is within these parameters is shortlisted (i.e. muscle boundary), as shown in Fig. 13 using the python program represented by Algorithm 9. The shortlisted line is then removed.

Algorithm 9 Shortlist Lines

```
BEGIN
FUNCTION shortlist_lines(lines):
    MIN_ANGLE <- 10
    MAX_ANGLE <- 70
    MIN_DIST  <- 5
    MAX_DIST  <- 256
    shortlisted_lines <- [x for x in lines IF
                              (x['dist']>= MIN_DIST) &
                              (x['dist']<= MAX_DIST) &
                             (x['angle']>= MIN_ANGLE) &
                              (x['angle']<= MAX_ANGLE)
                         ]
RETURN shortlisted_lines
END
```

D. IMAGE ENHANCEMENTS

It is not possible to remove the entire muscle area from some of the images because of the curved nature of the muscle boundary. Using the canny edge detection and Hough line transform on those images can remove most of the muscle, leaving a small portion along the muscle boundary. These remaining sections can be tackled by implementing effective image enhancements that can show ROIs and regions within the ROIs. These enhanced images can be used as training data for the D-CNN so that the D-CNN can learn to detect and extract the ROI (or mass regions) in the images faster and then use these ROIs to detect the cancerous parts. This approach can overcome the limitations of the Hough line transform for curved boundaries.

Image enhancements will make it easier to detect the details in a mammographic image. The enhancements are applied to the images using the ImageJ software tool called "Look Up Tables (LUTs)". Various LUTs were applied to the images to understand their behavior and select the best ones.

1) INVERT LUT

This LUT converts a mammographic image into an image that is similar to a photographic negative. LUTs can also be applied to grayscale images to produce pseudo-colored images, based on the progression of pixels in the image. Lehmann *et al.* [44] used the pseudo-coloring technique on medical X-ray images to produce enhanced visualization of diagnostic information.

For inversion, every gray level value **(V)** in the original image is replaced by **255-V**. When inverted, pixels with values 0 are converted to white and pixels with values 255 to black, as shown in Fig. 14.

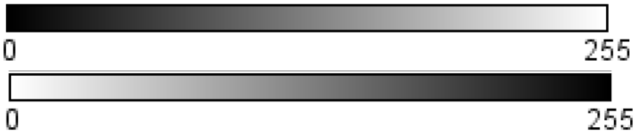


FIGURE 14. Invert LUT pixel value changes.

2) CTI_RAS LUT

This LUT is applied to show the ROIs in the images. Enhancing the images at the pre-processing stage with this LUT can save computational time for the D-CNN when extracting the ROIs and detecting the cancers.

As shown in Fig. 15, this LUT works by first inverting the gray level values in the original image (i.e., 0 = White and 255 = Black). Every gray level value **(V)** in the original image is replaced by **255-V**. After inverting the values, this LUT draws a rainbow-themed boundary based on the gray level value that is in the range (125, 255), as shown in Fig. 18, The range (125, 255) has been chosen because that is the range where the dark region starts and ends.

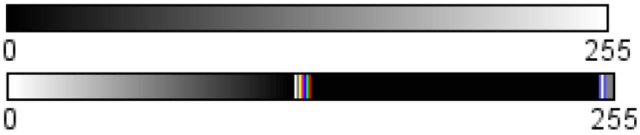


FIGURE 15. CTI_RAS LUT application process.

3) ISOCONTOUR LUT

This LUT is used to compose sets of isocontours within the ROIs, which can be used to extract features from different layers of the mass region in the mammographic image. As shown in Fig. 16, first the original image is enhanced with ISOCONTOUR LUT and then with the Invert LUT is applied to invert the image.

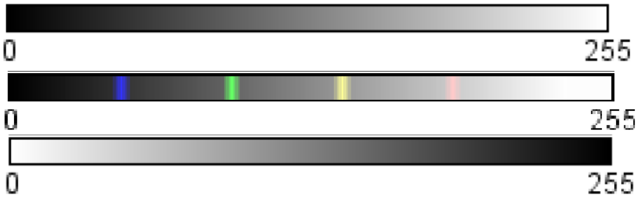


FIGURE 16. ISOCONTOUR LUT application process.

The ISOCONTOUR LUT draws contours based on the gray level values in the image using four colors (blue, green, yellow, red). A blue contour if the values are between (0, 50), green if the values are between (50,100), yellow in the range (100, 150), and red between (150, 200), where 0 is Black and 255 is White.

IV. RESULTS & ANALYSIS

A. BACKGROUND REMOVAL

Background removal has been applied to a total of 322 images from the MIAS dataset. The method has been implemented using python libraries. As can be observed from Table. 1,

TABLE 1. Background removal results.

Total Images	Successful	Accuracy
322	322	100%

the background (i.e., artifacts) is removed successfully from all images without affecting the pixel quality to a large extent. A histogram comparison is performed between the original images and the processed images to understand the changes in the images.

1) ROLLING BALL ALGORITHM

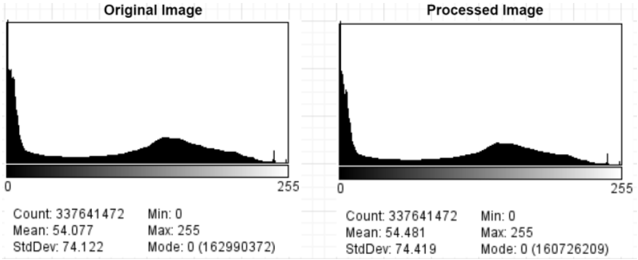
Histogram Analysis Between 'Original Image' & 'Processed Image Using "Rolling Ball Algorithm"'

The histogram analysis is performed using 322 images, by computing the **'mean gray value'** and the **'standard deviation'.** Gray values indicate the brightness of the pixels.

$$\text{Mean} = \frac{\textit{Sum of Gray Values of all the Pixels in Image}}{\textit{Total Number of Pixels in Image}} \tag{11}$$

Standard Deviation = Standard deviation in the Gray Values
Rolling Ball Radius = 5 Pixels
Total No. of Images = 322

The mean and the standard deviation are obtained by averaging the mean gray values and calculating the standard deviations of all the 322 images. As can be seen from Fig.17 and Table. 2, after processing the **original image,** the mean gray value has increased by **0.404,** which means that on average the brightness of the pixels in the images has been increased after applying the "rolling ball algorithm". The standard deviation has been increased by **0.297** after applying the "rolling ball algorithm".



Count: 337641472 Min: 0
Mean: 54.077 Max: 255
StdDev: 74.122 Mode: 0 (162990372)

Count: 337641472 Min: 0
Mean: 54.481 Max: 255
StdDev: 74.419 Mode: 0 (160726209)

FIGURE 17. Histogram analysis comparison.

TABLE 2. Mean & standard deviation value for 322 images.

Image	Mean	Std. Dev.
Processed Image	54.481	74.419
Original Image	54.077	74.122
Total Change	0.404	0.297

The mean and standard deviation values of all individual images (total 322) are plotted to understand the changes in individual images, see Fig. 18. Here, "blue" represents the mean and standard deviation of the 322 individual original images and "red" represents the values for the 322 individual rolling ball processed images.

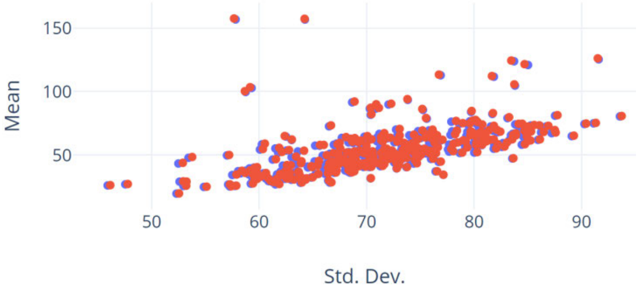


FIGURE 18. Histogram analysis comparison scatter plot for individual images (Total = 322).

a: MSE, PSNR, AND SSIM

After applying the rolling ball algorithm, the Mean Squared Error (MSE), the Peak Signal-to-Noise-Ration (PSNR), and the Structural Similarity Index Measure (SSIM) of the images were calculated to assess the quality of the image. The noise was removed from the images using the Rolling Ball algorithm. Noise can be added to the images at a later stage to create effective training data.

b: MSE

– Mean Squared Error

The MSE is the average squared difference between each pixel of the ground truth image (i.e., original image) and the processed image. An MSE which is close to 0 is considered better. If the MSE is equal to 0, there is no noise, and hence no need to find the PSNR.

$$\boldsymbol{MSE} = \frac{\mathbf{1}}{\boldsymbol{mn}} \sum\nolimits_{i=0}^{m-1} \sum\nolimits_{j=0}^{n-1} (\boldsymbol{G}(\boldsymbol{i},\boldsymbol{j}) - \boldsymbol{P}(\boldsymbol{i},\boldsymbol{j}))^{\mathbf{2}} \tag{12}$$

where, **G** is the ground truth image (i.e., original image) and **P** is the processed image. **m** and **n** represent the pixels of **G** and **P** and **i, j** represents the rows of the pixels **m, n.**

c: PSNR

– Peak Signal-to-Noise-Ration

PSNR is the ratio between the maximum possible power of a signal and the power of the corrupting noise affecting the quality of the image. It is calculated, using the previously calculated MSE value.

$$\boldsymbol{PSNR} = 20\log_{10}\left(\frac{(\boldsymbol{MAX})}{\sqrt{\boldsymbol{MSE}}}\right) \tag{13}$$

where, **MAX** is the maximum pixel value of the image (i.e.,255). For an 8-bit image, the PSNR values are usually between **30 to 50 dB** [49]. If the PSNR value for an image is high, the image quality is considered good. Values over **40 dB** are considered very good, whereas below **20 dB** are considered unacceptable [49].

d: SSIM

– Structural Similarity Index Measure

SSIM is used to measure image quality degradation caused by processing. SSIM is calculated between 2 images (Original Image and the Processed Image). The index is in the range of -1 to 1, where 1 indicates 'perfect structural similarity' and 0 'no similarity'. It is calculated by sliding a 'GAUSSIAN' window of size 11 × 11 [29].

$$SSIM\,(x,y)=\frac{\left(2\mu_x\mu_y+c_1\right)\left(2\sigma_{xy}+c_2\right)}{\left(\mu_x^2+\mu_y^2+c_1\right)\left(\sigma_x^2+\sigma_y^2+c_2\right)} \tag{14}$$

where, $\boldsymbol{\mu_x}$ and $\boldsymbol{\mu_y}$ is the averages of two images $(\mathbf{x},\ \mathbf{y})$ calculated by using the gaussian window. $\boldsymbol{\sigma_x^2}$ and $\boldsymbol{\sigma_y^2}$ is the variance and $\boldsymbol{\sigma_{xy}}$ is the covariance of $\mathbf{x}$ and $\mathbf{y}$. $\mathbf{c_1}$ and $\mathbf{c_2}$ are the two variables used to stabilize the division, where $\mathbf{c_1} = \mathbf{(0.01 \times 255)^2}$ and $\mathbf{c_2} = \mathbf{(0.03 \times 255)^2}$.The values 0.01 and 0.03 are by default.

MSE, PSNR, and SSIM values are calculated by comparing the ground truth **(G)**(i.e. original images) and the rolling ball algorithm processed images **(P)**. Values for 10 images that were selected at random are shown in Table 3.

TABLE 3. MSE, PSNR, SSIM values for 10 images.

Image (G - P)	MSE	PSNR	SSIM
mdb001	1.8495	45.4601	0.9852
mdb018	1.4440	46.5348	0.9880
mdb022	2.3885	44.3495	0.9804
mdb033	1.6491	45.9582	0.9871
mdb049	2.1819	44.7424	0.9816
mdb086	1.9967	45.1276	0.9834
mdb095	3.1219	43.1864	0.9763
mdb137	2.9069	43.4963	0.9799
mdb166	1.4729	46.4488	0.9873
mdb300	2.7332	43.7640	0.9702

PSNR values for all 322-rolling ball processed images are in the range of **42-47 dB**, which suggests that the images are of good quality [49]. SSIM values are in the range of **0.97** to **0.99**, which means the images are structurally similar even after the application of the rolling ball algorithm.

2) HUANG's FUZZY THRESHOLDING & MORPHOLOGICAL TRANSFORMATIONS

After processing the rolling ball processed images with Huang's fuzzy thresholding and morphological transformations, the results were analyzed.

Histogram Analysis Between Original Image and Background Removed Image

The background is removed from 322 images, and the mean gray value and standard deviation were calculated for all the images.

It can be seen from the histograms above that there is a difference in the histograms of the original image histogram and the background removed histogram.

TABLE 4. Mean & standard deviation comparison.

Image	Mean	Std. Dev.
Original Image	54.077	74.122
Background Removed Image	51.100	74.337
Total Change	2.977	0.215

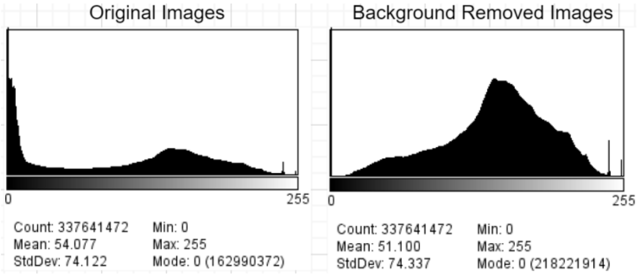


FIGURE 19. Histogram analysis (322 Images).

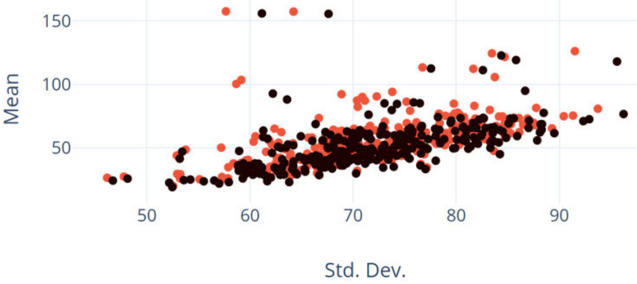


FIGURE 20. Histogram analysis comparison scatter plot for individual images (Total = 322).

As can be observed from Table. 4, after removing the background, the total change (original image – background removed image) in the mean gray value of 322 images has decreased by **2.977** and the standard deviation has increased by **0.215**. This has increased the contrast of the images, making the details clearer [50].

The values of the mean and standard deviations values of individual images (total 322) were plotted to understand the changes in individual images (see Fig. 20). Here, "red" represents the mean and standard deviation of 322 individual original images and "black" represents 322 individual background removed images.

Images After Applying Rolling Ball, Huang's Threshold & Morphological Transformations

As can be seen in Fig. 21, the proposed method for background removal is capable of removing artifacts from the image, along with unwanted areas. Removing unwanted areas at the pre-processing stage will save computational time for the D-CNN.

3) COMPARISION WITH RESULTS FROM OTHER RESEARCH

Hazarika and Mahanta [19] used morphological transformations to remove the background from mammogram images. They used a structuring element of length 10 pixels and angel 15 to apply a closing operation and a disk structuring element of radius 2 pixels for the erosion operation, which

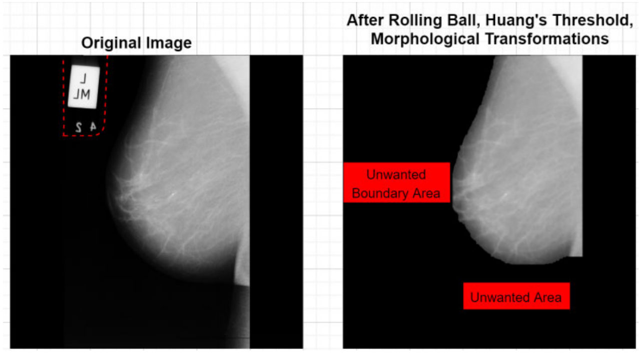

Original Image
After Rolling Ball, Huang's Threshold, Morphological Transformations
Unwanted Boundary Area
Unwanted Area

FIGURE 21. Final images after background removal process.

TABLE 5. Results comparison.

Authors	Methods Used	Dataset	Results
Manasi et al. (2018) [19]	Binary Thresholding & Morphological Operations	322 MIAS Images	98.7%
Mina et al. (2015) [51]	Binary Thresholding & Morphological Operations using MATLAB	322 MIAS Images	99.06%
Proposed Method	**Rolling Ball Algorithm, Huang's Fuzzy Threshold, Morphological Transformation**	**322 MIAS Images**	**100%**

resulted in an accuracy of 98.7% in removing the artifacts. Mina *et al.*, [51] also used "morphological transformations" to remove the background from the images. They have used a flat disk-shaped structuring element (STREL) of radius 5 pixels. The operations were implemented using MATLAB functions and achieved an accuracy of 99.06%. Although both Hazarika and Mahanta [19] and Mina and Isa [51] used the same methods (binary thresholding and morphological transformations), their results are different, as can be seen from Table. 5, and they did not provide results in terms of effects of processing on the image quality and details. Our research has also used morphological transformations, but along with the rolling ball algorithm we implemented Huang's fuzzy thresholding, achieving a success rate of 100% in removing the background from the images without affecting the image quality and details. Implementing the methods proposed by [19] and [51] through OpenCV Python yielded an average computational time of 0.008 seconds in processing each image, whereas the average computational time for our proposed method (Rolling ball algorithm, Huang's fuzzy thresholding, and Morphological transformations) was only 0.045 seconds.

TABLE 6. Mean & standard deviation.

Image	Mean	Std. Dev.
Original Image	54.077	74.122
Background Removed Image	51.100	74.337
Resized & Right-Side Flipped Images	51.099	74.169

TABLE 7. Pectoral muscle removal results.

Total Images	Successful	Result
322	319	99.06%

B. PECTORAL MUSCLE REMOVAL

A total of 322 background removed images are used for implementing the proposed method for pectoral muscle removal. Firstly, the 322 images are downsized from 1024 × 1024 pixels to 256 × 256 pixels. Secondly, the resized images are flipped to the right side. Finally, the pectoral muscle is removed. The images are downsized so that the computations can be performed faster.

Histogram Analysis After Resizing the Right-Side Flipped Image

To understand how the resizing and flipping affect the quality of the details in the image, the mean and standard deviation is calculated for 322 resized and flipped images, see Table 6.

The mean gray value has remained approximately the same compared to the value of the background removed image. It is lower than the mean of the original image by 2.978. The standard deviation has decreased by 0.168 compared to the background removed image value but is still higher than the original image value.

After implementing the proposed method for pectoral muscle removal on 322 MIAS resized and right-side flipped images, most of the muscle area could be removed from 319 or 99.06% of the images, as shown in Table. 7.

Dice Similarity Coefficient

In this research, Dice similarity is calculated to estimate the accuracy of breast border and muscle boundary estimation or extraction.

$$DSC = \left(\frac{2 \, * \, Area\ of\ Overlap}{Total\ Number\ of\ Pixels\ in\ both\ Images} \right) \quad (15)$$

The area of overlap is estimated between the 'breast and muscle boundary extracted from original image' and 'boundary extracted from processed image', as seen in Fig. 23. The Dice coefficient is calculated between two binary images that have borders extracted. The Dice coefficient values are in the range of 0 to 1, where 1 indicates a high similarity between the images.

As seen in Table. 8, the same 10 images that were used to calculate MSE, PSNR, and SSIM values are used to calculate the dice coefficient score. Averaging, the above values produced a Dice Score of **0.977± 0.015 or 97.7±1.5%**,

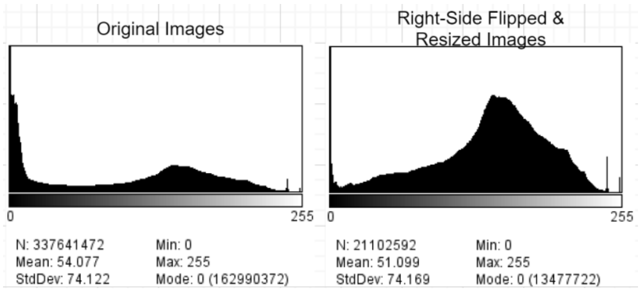


FIGURE 22. Histogram comparison.

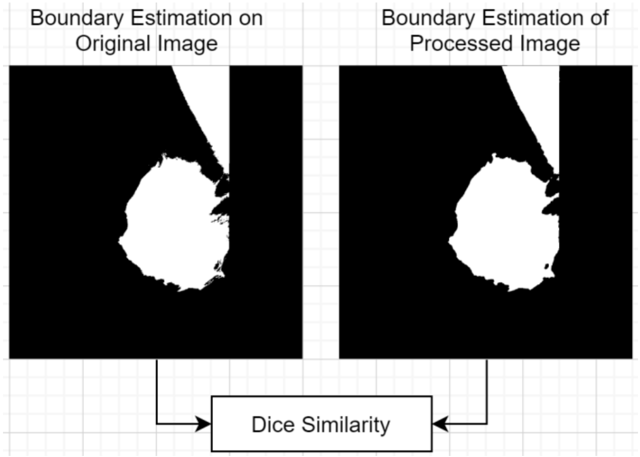

Boundary Estimation on Original Image
Boundary Estimation of Processed Image
Dice Similarity

FIGURE 23. Boundary estimation.

TABLE 8. DSC (Ground truth and processed image).

Image	DSC
mdb001	0.9920
mdb018	0.9862
mdb022	0.9763
mdb033	0.9738
mdb049	0.9688
mdb086	0.9672
mdb095	0.9747
mdb137	0.9829
mdb166	0.9633
mdb301	0.9886

which suggests that applying the proposed methods on the mammogram images did not affect the breast boundary and pectoral muscle boundary estimation. Therefore, the Canny edge detection process will be able to detect the muscle boundary accurately and remove the pectoral muscle. The table below compares the dice scores of various researchers.

TABLE 9. Dice similarity coefficient comparison.

Authors	Dice Score
Ali, M.J. et al. (2020) [65]	96.16%
Rampun, A. et al. (2019) [66]	97.5%
Taghanaki, S.A., et al. (2017) [26]	97.8±0.8%%
Rampun, A. et al. (2017) [22]	97.8%
Tzikopoulos, S.D. et al. (2011) [64]	94.5%
Proposed Method	97.7±1.5%

1) MSE, PSNR AFTER MUSCLE REMOVAL

During the pectoral muscle removal process noise is added to the background removed images. Often neural network models perform well at the training stage and testing stage, but poorly when used in the real world. Using clean data for training and then using real-world data (which is usually less clean) can decrease the performance of the neural network.

After processing the background removed images with pectoral muscle removal methods, some noise is added to the images with the intention that the image quality is not affected too much. MSE and PSNR values of the original and the processed images are calculated.

For all the 322 images the values of PSNR stayed between 35-43 dB, which is considered acceptable as it is in the range of 30-50 dB [49]. Results were collected using the same 10 images that were used before, as shown in Table. 10.

TABLE 10. MSE & PSNR.

Image	MSE	PSNR
mdb001	8.3762	38.9003
mdb018	6.2108	40.1992
mdb022	8.3826	38.8970
mdb033	9.6968	38.2644
mdb049	8.8646	38.6542
mdb086	8.2119	38.9863
mdb095	7.1120	39.6108
mdb137	5.7179	40.5583
mdb166	4.0736	42.0309
mdb301	9.0002	38.5870

2) COMPARISON OF RESULTS WITH PREVIOUS RESEARCH

Taghanaki *et al.* [26] proposed methods for prefiltering and identification of the breast boundary and extraction of muscle using geometric rules, and optimization methods for images with curved muscle boundaries. They applied Contrast-Limited Adaptive Histogram Optimization (CLAHE) parameters to enhance the images to extract clear details and strong edges. Then, they applied "Canny edge detection" to images, converted to binary, to extract the breast boundary. To identify the location of the muscle, a vector distance transformation strategy was implemented to search for the medial axis, radius, and center of the muscle boundary, optimizing the methods for images with curved muscle boundaries. They applied their methods on 322 MIAS images and achieved an average Dice score of 97.8±0.8% for extraction of the muscle region. This resulted in an

TABLE 11. Results comparison.

Authors	Method	Dataset	Results
Rahimeto, S. et al. (2019) [62]	Connected Component Labeling & Otsu Binary Thresholding	322 MIAS + Locally Collected Images	93.36%
Taghanaki, S.A., et al., (2017) [26]	Geometry-based Segmentation using Canny Edge Detection	Random Images from INBreast [54] IRMA [55] MIAS [7] 872 Images	95%
Vikhe, P.S. et al. (2017) [40]	Enhancement Filter & Least Square Error (LSE)	322 MIAS Images	96.56% (Acceptable)
Bora, V.B. et al (2016) [31]	Texture Gradient, Hough Line Transform, Euclidean Distance Regression with Polynomial Modelling for Curve Fitting	340 Images (200 MIAS, 100 Computed Radiography Images, 40 Full-Field Digital Images)	96.75%
Maitra, I.K., et al. (2012) [52]	Seeded Region Growing Algorithm	322 MIAS Images	95.71% (Acceptable)
Bick, Ulrich, et al [53] (1995)	Binary Thresholding, Region Growing, Morphological Filtering	740 Digitalized Mammograms	97% (Acceptable)
Proposed Method	**Canny Edge Detection, Hough Line Transform**	**322 MIAS Images**	**99.06%**

accuracy of 95% in removing the muscle. In our research, after processing the images with "rolling ball algorithm" and "Huang's fuzzy thresholding" to enhance the details and edges in the images, using the Canny edge detection and Hough line transform produced a dice score of 97.7±1.5% in estimating the pectoral muscle boundary and a 99.06% accuracy in removing the muscle. For images with curved muscle boundaries, the remainder after application of these removal methods is addressed by implementation of LUTs (image enhancements) to highlight the ROIs and regions within the ROIs. This saves computational time for the muscle removal process, as shown in Table 20. Rampun *et al.* [22] proposed methods for estimation of breast and muscle boundary, and segmentation of the muscle region. They investigated 25 image features and selected "entropy" because of its simplicity to distinguish texture along the skin-air breast boundary. They then used Canny edge detection and active contour-based methods for breast boundary estimation. They used the Robust Local Regression MATLAB function to smooth the detected muscle boundary. They obtained a dice similarity coefficient of 97.8% and an accuracy of 99.4% in detecting and estimating the pectoral muscle boundary. The only drawback of their method is the average computational time, which is 7 seconds (see Table. 20) per image, compared to 0.35 seconds for our method. Bora *et al.* [31] proposed a method for pectoral muscle removal, based on texture gradient and Hough line transform along with Euclidean distance regression with polynomial modeling for curve fitting. They were able to achieve 96.75% accuracy. The drawback of their method is the computational time, which is 4.81 seconds because they tried to focus on removing the curved muscle boundaries in the processing stage itself. We believe that rather than focusing on curved boundaries in the processing stage itself, which increases the computation time, this can be tackled using the LUTs to extract the ROIs and regions within the ROIs. Rahimeto*et al.*, [62] in their research proposed an automatic pectoral muscle removal method based on the connected component labeling method. The muscle region was extracted using Otsu's multi-thresholding method. They were able to achieve 93.36% accuracy in detecting the pectoral muscle. The average computational time of their muscle removal method was 1.3314 seconds per image, which is high when compared to the computational time of our muscle removal method (i.e., 0.35 seconds).

Our research has used both Canny edge detection and Hough line transform after processing the images with background removal methods, to detect and remove the muscle area from the images. We achieved a dice similarity of 97.7±1.5% in estimating the muscle boundary and an accuracy of 99.06% in removing the muscle. The computational time is only 0.35 seconds which is better most of the existing methods. It is important to understand that the primary goal of this research is to remove as much unwanted area as possible from the mammographic images using computationally simple image pre-processing methods. While some images have small portions of the pectoral muscle left after applying the proposed method, this can be resolved by using image enhancements that can highlight and extract the ROIs and regions within the ROIs.

C. IMAGE ENHANCEMENTS

First, the gray-scale images are converted to RGB color images to collect the RGB values in the image. Applying the LUTs on the images can extract the ROIs and regions within the ROI.

1) INVERT LUT

a: HISTOGRAM ANALYSIS BETWEEN ORIGINAL IMAGE & INVERT LUT

As observed in Fig. 24, the mean gray value and standard deviation value stayed the same even after inverting the gray level values in the image. Image "**mdb001**" is used for the analysis. Applying the LUT did not affect the image.

b: RGB ANALYSIS

RGB (Red, Green, Blue) analysis is performed to understand the change in intensity values of each pixel in the LUT applied

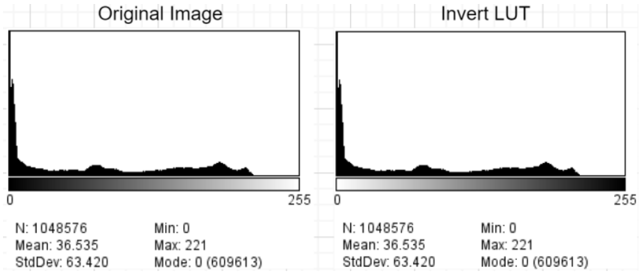


FIGURE 24. Histogram comparison.

TABLE 12. RGB analysis original image.

Label	Mean	Std. Dev.
Red	36.535	63.420
Green	36.535	63.420
Blue	36.535	63.420
(R+G+B)/3	36.535	63.420

TABLE 13. RGB analysis after applying invert LUT.

Label	Mean	Std. Dev.
Red	218.465	63.420
Green	218.465	63.420
Blue	218.465	63.420
(R+G+B)/3	218.465	63.420

image. First, the 8-bit grayscale image is applied with a LUT and then converted to an RGB color image.

RGB Analysis of Original Image (8-bit image converted to RGB color image):

As can be seen from Table. 12, the red, green, and blue means values in the image stayed neutral. When all the colors are equal, it indicates that there is a neutral color in the original image, such as white, gray, or black, (in this class a gray-scale image).

As observed in Table. 12 and Table. 13, the values remain constant for both images. Image "**mdb001**" used for analysis.

Here, the veins and other details in the image can be more easily identified. As can be seen in Fig. 25, after applying the invert LUT, the mean gray value has also inverted (**255-V**) i.e., 218.465.

2) CTI_RAS LUT

a: HISTOGRAM ANALYSIS BETWEEN ORIGINAL IMAGE & CTI_RAS LUT

As observed in Fig. 26, the mean and standard deviation values remain the same for both images. Image "**mdb001**" is used for the analysis. Applying the LUT did not affect the image.

b: RGB ANALYSIS

Between Original Image (Table. 11) and CTI_RAS Applied Image (Table. 14).

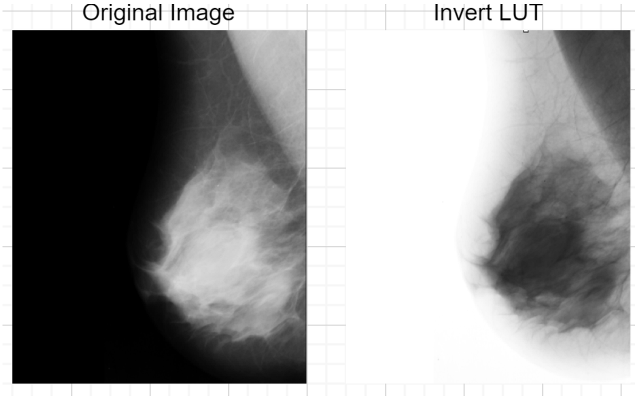


FIGURE 25. Invert LUT.

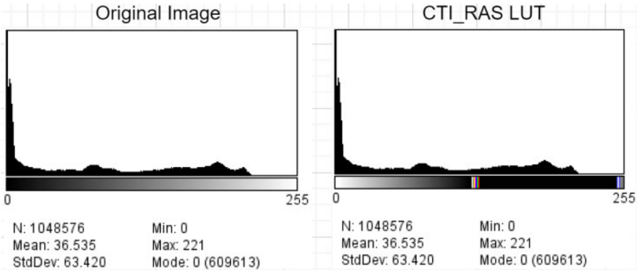


FIGURE 26. Histogram comparison.

TABLE 14. RGB analysis after applying CTI_RAS LUT.

Label	Mean	Std. Dev.
Red	195.361	97.232
Green	195.103	97.412
Blue	195.109	97.407
(R+G+B)/3	195.191	96.959

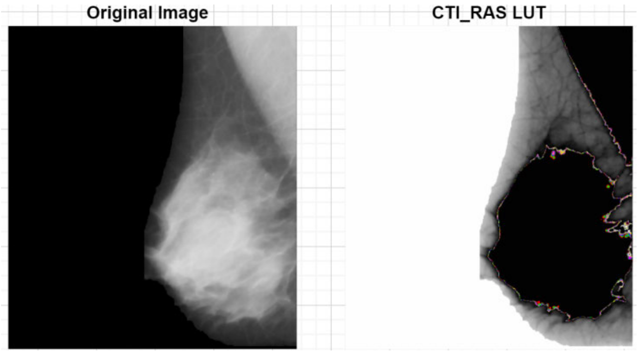


FIGURE 27. CTI_RAS LUT application.

RGB analysis of CTI_RAS LUT Image (8-bit image applied with LUT, then converted to RGB color):

As observed in Table. 14, RGB analysis after applying the CTI_RAS LUT produced slightly different values that are close to each other, which means the corresponding color in the original image is closer. As seen in Fig. 27, this change in values will be used by the D-CNN to detect the borders of the ROI in the image and then extract it for detecting cancer. Image "**mdb001**" used for analysis.

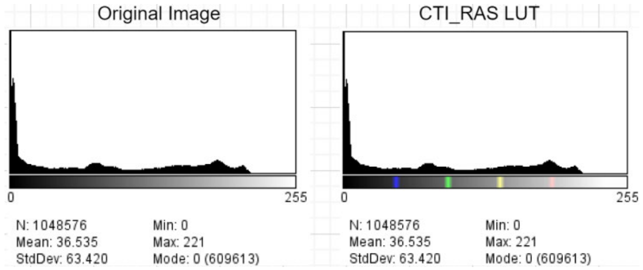


FIGURE 28. **Histogram comparison.**

TABLE 15. **RGB analysis after applying ISOCONTOUR LUT.**

Label	Mean	Std. Dev.
Red	41.052	70.784
Green	41.342	70.715
Blue	40.853	69.142
(R+G+B)/3	41.080	69.742

TABLE 16. **RGB analysis after applying invert LUT.**

Label	Mean	Std. Dev.
Red	221.322	60.163
Green	221.085	58.983
Blue	221.994	58.175
(R+G+B)/3	221.800	58.826

3) ISOCONTOUR LUT

a: HISTOGRAM ANALYSIS BETWEEN ORIGINAL IMAGE & ISOCONTOUR LUT

As observed in Fig. 28, the mean and standard deviation values remain the same for all the images. Image "**mdb001**" is used for the analysis. Applying the LUT did not affect the image.

b: RGB ANALYSIS

A comparison between the Original Image (Table. 12) and the ISOCONTOUR Applied Image (Table 15) and the Invert LUT Applied Image (Table 16) was done.

RGB analysis of ISOCONTOUR LUT image (ISOCONTOUR LUT applied on the original image, and then converted to RGB color) is shown in Table 15:

RGB analysis of Invert LUT image (ISOCONTOUR LUT image applied with Invert LUT, then converted to RGB color) is shown in Table 16.

As can be seen in Fig. 29, applying invert LUT on the ISOCONTOUR LUT image has resulted in identifying more regions in the image.

RGB analysis shows the change in values. The more they differ, the stronger and purer the color in the image, as seen in Fig. 29. These changes in RGB values will be used by D-CNN to extract features within the ROIs. Image "**mdb001**" was again used for analysis.

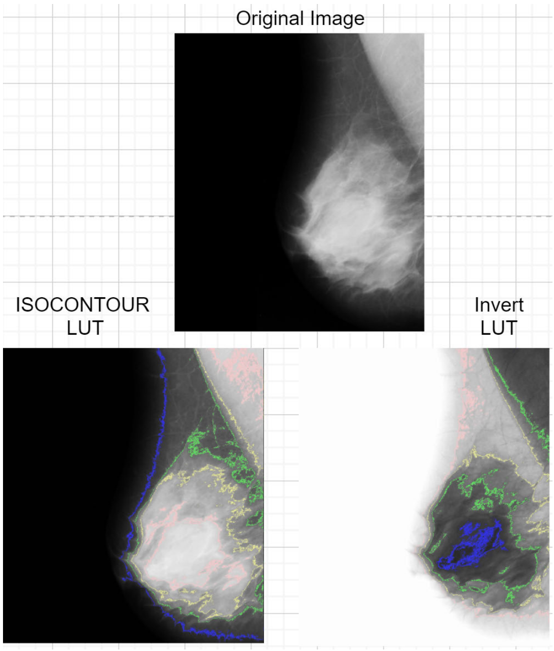


FIGURE 29. **ISOCONTOUR LUT application.**

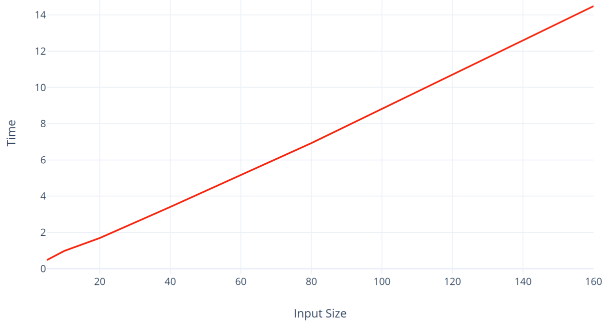


FIGURE 30. **Time complexity graph (Rolling ball).**

D. COMPLEXITY OF THE ALGORITHMS & PROGRAMS

Both time complexity (BIG O) and cyclomatic complexity (McCabe's) are calculated for the algorithms and programs used in this research. Time complexity is calculated by measuring the run-time of the algorithms against inputs of various sizes. Cyclomatic complexity is used to measure the complexity of the algorithms and the python code programs involved in this research.

1) TIME COMPLEXITY

Time complexity describes the amount of time required to run the algorithms and programs used in this research. The MIAS dataset (322 images) is divided into random inputs of various sizes for the algorithms and programs so that the execution time can be plotted and represented using the 'Big O notation'. Overall time complexity is calculated for the background removal process, pectoral muscle removal process, and the image enhancement process.

a: BACKGROUND REMOVAL

Time complexity is calculated separately for the execution of the rolling ball algorithm, Huang's fuzzy thresholding, and morphological transformation. As shown in Table 17, various input sizes are used to calculate the execution time.

TABLE 17. **Rolling ball algorithm time complexity.**

Input Size	Execution Time (Seconds)
5	0.474
10	0.984
20	1.691
40	3.41
80	6.931
160	14.488

TABLE 18. **Time complexity (Huang's thresholding + morphological transformations).**

Input Size	Execution Time (Seconds)
10	0.24
20	0.38
40	0.68
80	1.22
160	2.15
320	3.67

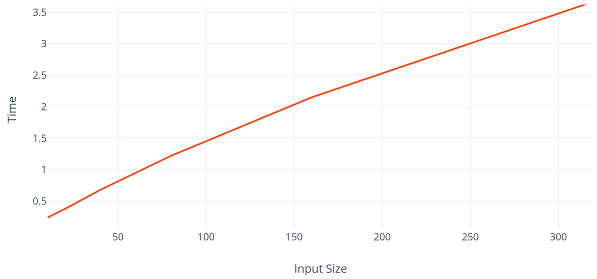


FIGURE 31. **Time complexity graph (Huang's thresholding + morphological transformation).**

For Rolling Ball Algorithm,

Based on the values in Table 17 a graph is plotted with the x-axis representing the input size and the y-axis representing the execution time, as seen in Fig. 30.

It can be seen from Fig. 30 that the execution time increases linearly and in direct proportion to the input sizes. Therefore, the time complexity is **O(n)** for Huang's Fuzzy Thresholding and Morphological Transformation,

Based on the values from Table 18, an (x, y) graph is plotted of input size versus execution time, as shown in Fig. 31. It can be seen that the graph is increasing linearly, corresponding to the time complexity of **O(n).**

b: PECTORAL MUSCLE REMOVAL

The time complexity is also calculated for the execution (run-time) of the pectoral muscle removal python program (Canny edge detection and Hough line transform) for inputs of various sizes.

Based on values from Table 19, a graph is plotted of input size versus execution time in seconds, see Fig. 32. It can be noted that the graph is increasing linearly. Therefore, the time complexity is **O(n).**

The total computational time of our method is shorter than that of most other methods [62], [67], [68], [69], as shown in Table 20.

TABLE 19. **Time complexity (Pectoral muscle removal process).**

Input Size	Execution Time (Seconds)
1	0.0268
4	1.421
8	2.7575
16	5.674
32	11.5234
64	21.4004

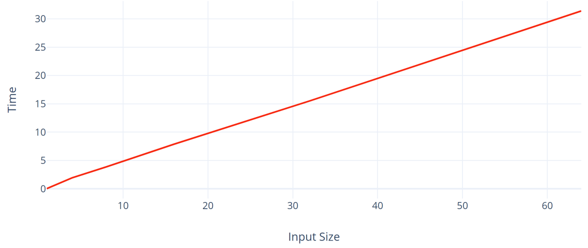


FIGURE 32. **Time complexity graph (Pectoral muscle removal process).**

TABLE 20. **Computational Time Comparison (Pectoral Muscle Removal Process).**

Authors	Computational Time (s)
Rahimeto, S. et al. (2019) [62]	1.3314
Rampun, A. et al. (2017) [22]	7
Bora, V.B. et al. (2016) [31]	4.81
Chen, C. et al. (2015) [68]	0.3684
Vaidehi, K. et al. (2013) [67]	0.5450
Proposed	**0.35**

c: IMAGE ENHANCEMENTS

Time complexity is calculated for execution (run-time) of LUTs on different input sizes. The execution time is constant for all the input sizes. For an input size of 320 images, it took 0.15 seconds to apply the LUT and the same time was found for input sizes of 1, 10, and 50 images. Therefore, the time complexity is **O(1)**.

2) CYCLOMATIC COMPLEXITY

Cyclomatic complexity is the number of decisions a block (function, method, or class) of code contains. This number is called the McCabe number and is equal to the number of linearly independent paths through the code (based on the control flow graph). To compute the cyclomatic complexity, we have used a python tool called "**Radon**". The complexity is ranked from A to F based on the score, where 'A' (1-5) denotes the most simple and best code, 'B' (6-10) denotes well-structured and stable blocks, 'C' (11-20) denotes moderate and slightly complex block and 'F' (41+) denotes very high risk and unstable code. Cyclomatic complexity is calculated by using the equation $\mathbf{M = E - N + 2P}$, where **E** is the number of edges in the control flow graph of the program, **N** is the number of nodes in the graph and **P** is the number of connected components.

TABLE 21. Overall cyclomatic complexity.

Algorithm or Program	Rank	Average Score
Rolling Ball Algorithm	B	6.2
Huang's Thresholding	C	11
Morphological Transformations	A	2
Pectoral Muscle Removal	A	2
Total Average of Average Score		**5.3 (A)**

As shown in Table 21, the overall average cyclomatic complexity score for the combined algorithms and programs involved in this research for image pre-processing is equal to 5.3 (A), which means they are the simplest and best.

E. STATISTICAL SIGNIFICANCE OF THE RESULTS

Statistical significance quantifies whether a result is likely to be due to chance or due to the factor of interest [59]. The results of background removal and the results of pectoral muscle removal are evaluated separately.

1) BACKGROUND REMOVAL RESULTS

After removing the background from the original images using the proposed method, the results are generated in terms of histograms. Evaluation of the significance of the results is based on the histogram values of the background removed images. As the artifacts and noise have been removed from all the images, the mean and standard deviation of the pixel values should be in the same range for all the images. Evaluating the significance of the results is a step-by-step process. Two datasets are created with histogram results of 31 random background removed images each.

Firstly, a "Null Hypothesis" is created by stating that there is no difference between the datasets. Secondly, an "Alternative Hypothesis" stating the opposite of the null hypothesis is created. These hypotheses are used to determine the significance of the results.

Determining the 'Alpha' value is an important step to determine the significance level, which is the probability of rejecting a null hypothesis when the hypothesis is true. Here, **alpha** (α) = **5% or 0.05**. Based on the alpha value, the confidence level is calculated (i.e., **(1 –**α)), which is equal to **0.95 or 95%.**

a: STANDARD DEVIATION

After determining the alpha value, "Standard Deviation" values (S1 and S2) for both the datasets are calculated using equation 16.

$$S = \sqrt{\frac{\sum (x_i-\mu)^2}{(N-1)}} \tag{16}$$

where **S** is the standard deviation, $\mathbf{x_i}$ is individual data and μ is mean of the dataset, **N** is the size of the dataset (i.e., **31**). Calculating **S1** and **S2** for 2 datasets of size **N = 31** using equation 16 yielded the values **10.295** and **13.380,** respectively.

b: STANDARD ERROR

The standard error is calculated based on the standard deviation values of two datasets of size 31, using equation 17.

$$S_d = \sqrt{(\frac{S_1}{N_1}) + (\frac{S_2}{N_2})} \tag{17}$$

where S_d is the standard error and S_1 and S_2 are standard deviations of two datasets of size N_1 and N_2. Calculating S_d using equation 17 yields **0.8734.**

c: T-SCORE

T-score is used to compare two datasets to identify the probability that they are significantly different.

$$T = \frac{(\mu_1 - \mu_2)}{S_d} \tag{18}$$

where T is T-score, μ_1 and μ_2 is mean of dataset 1 and dataset 2 and S_d is the standard error. Calculating the T-score using equation 18 yields **15.539.**

d: DEGREE OF FREEDOM

The degree of freedom ($\mathbf{d_f}$) tells about how many values in a calculation can vary acceptably. It is calculated by adding the two datasets and subtracting 2. Calculating $\mathbf{d_f}$ yields **60**(i.e., ((31 + 31) – 2)).

e: T-DISTRIBUTION TABLE TO FIND STATISTICAL SIGNIFICANCE

Using the T-distribution table values, the significance of the conclusion is determined. Using $\mathbf{d_f}$ and T-score values mentioned above gives a **p-value** is smaller than **0.0005** (i.e., confidence level greater than **0.9995**), which is well below the significance level (α) of **0.05**. Therefore, it can be concluded that the background removal results are statistically significant.

2) PECTORAL MUSCLE REMOVAL RESULTS

The statistical significance of the pectoral muscle removal results is calculated using "Dice-Similarity Co-efficient" values because the muscle removal process depends on muscle boundary estimation and extraction. The significance of the results is determined by following the step-by-step process explained before.

Here, the 'Alpha (α) value' is set as **0.05 or 5%**. Therefore, the confidence level is **0.95 or 95%.** Two datasets of size **N = 31** are selected, which contains the dice scores from random images.

Calculating the standard deviation (**S**) for the datasets using equation 16 yields **0.0078** and **0.0167**, respectively. Using the **S** values of both the datasets, the standard error ($\mathbf{S_d}$) is calculated using equation 17, which yields **0.0328**.

Using the mean values of both the datasets (**0.9808** and **0.9321)** and the standard error value, the T-score is calculated using equation 18 which yields **1.7294**. The degree of

TABLE 22. Significance of results (Background removal results & pectoral muscle removal results).

Significance of Results	Dataset Size (N)	Standard Deviation (S)	S_d	d_f	P-Value
Background Removal Results	31	10.295 (S1)	0.8734	60	<0.0005
	31	13.380 (S2)			
Pectoral Muscle Removal Results	31	0.0078 (S1)	0.0328	60	0.05-0.025
	31	0.0167 (S2)			

freedom (**d_f**) is equal to 60. Using **d_f** and T-score values, the T-Distribution table is checked to determine the significance of the results. In this case, the **p-value** is between the significance level (α) of **0.05 and 0.025** (i.e., the confidence level is between **0.95 and 0.975**). It can therefore be concluded that the pectoral muscle removal results are statistically significant. Table 22 illustrates the statistical significance of results for background removal and pectoral muscle removal results.

V. CONCLUSION

Developing effective training data for the D-CNN is the primary goal of this research. Therefore, methods for background removal, pectoral muscle removal, and image enhancements are proposed. During the background removal process, artifacts and noise are removed from the images. During the pectoral muscle removal process, noise is again added into the images without affecting the quality of the details in the images, so that data can represent real-world scenarios while training the D-CNN. This may improve the performance of the neural network when used in the real world. LUTs are applied during the image enhancement process to outline the ROIs and regions within the ROIs. This research has implemented the "Rolling ball algorithm", "Huang's Fuzzy Thresholding", "Morphological Transformations", "Canny Edge Detection", "Hough Line Transform" and "Look Up Tables (LUTs)" on 322 MIAS images to create datasets for D-CNN. The proposed methods can remove background from 100% of the images and pectoral muscle from 99.06% of the images and the image-enhancements can outline the ROIs and regions within the ROIs clearly.

FUTURE WORK

After processing the mammographic images using the proposed methods, training data, validation, and testing data for the Deep-CNN will be created. As shown in Fig.33 a Deep-CNN will be built and trained to detect and classify the abnormalities and their severities from the mammogram images, along with various other characteristics.

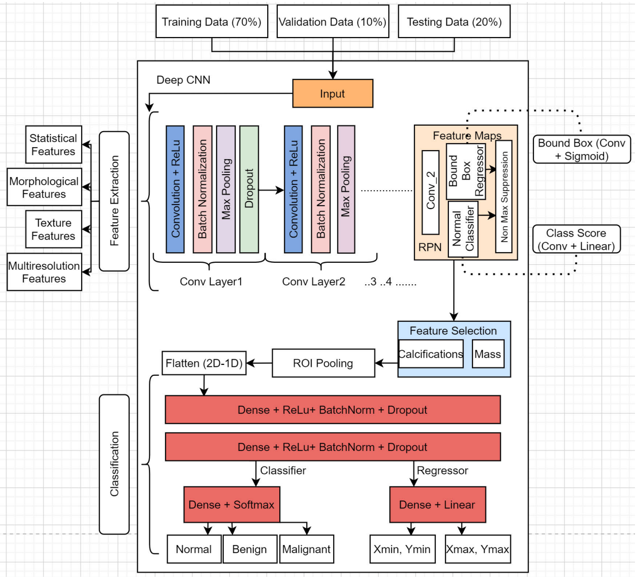


FIGURE 33. D-CNN framework.

The processed images will be separated into three datasets with 70% for training, 10% for validation, and 20% for testing. First, the training data will be passed through several convolutional layers to apply the Rectified Linear Unit (ReLU) operation to increase non-linearity in the images, to perform batch normalization and max polling, and finally, dropout will be used to improve the generalization performance of the D-CNN model. Feature maps are used at each layer to understand what features each layer has detected from the passage of the image. Several Dense ReLU layers will be used to improve the performance of the trained neural network. Various features such as statistical features, morphological features, texture features, multi-resolution features, etc., will be extracted from the passage of images through the convolution layers. The extracted features will be used to classify the images into normal, benign, and malignant cases. The neural network will also be trained to identify the coordinates of the abnormality. During the training process, validation data will be used to fine-tune the parameters and make changes to the network accordingly, so that the performance can be improved. Finally, testing data will be used to test the trained network.

SOURCE CODES

To facilitate reproducible research and help the academic community we are releasing the python source codes for the algorithms and programs that are used in this research. The source codes are released on: https://github.com/ABRB13/BreastCancer

REFERENCES

[1] F. Z. Francies, R. Hull, R. Khanyile, and Z. Dlamini, "Breast cancer in low-middle income countries: Abnormality in splicing and lack of targeted treatment options," *Amer. J. Cancer Res.*, vol. 10, no. 5, p. 1568, 2020.

[2] Australian Institute of Health and Welfare. (2019). *Cancer Data In Australia, Cancer Rankings Data Visualization*. Accessed: Dec. 7, 2020. [Online]. Available: https://www.aihw.gov.au/reports/cancer/cancer-data-in-australia/contents/cancer-rankings-data-visualisation

[3] D. Saslow, C. Boetes, W. Burke, S. Harms, M. O. Leach, C. D. Lehman, E. Morris, E. Pisano, M. Schnall, S. Sener, R. A. Smith, E. Warner, M. Yaffe, K. S. Andrews, and C. A. Russell, ''American cancer society guidelines for breast screening with MRI as an adjunct to mammography,'' *CA A, Cancer J. Clinicians*, vol. 57, no. 2, pp. 75–89, Mar. 2007.

[4] Healthdirect.gov.au. (2020). *Mammography*. Accessed: Dec. 7, 2020. [Online]. Available: https://www.healthdirect.gov.au/mammography#:~:text=What%20does%20a%20mammogram%20cost,a%20rebate%20on%20those%20tests

[5] G. Jobs. (2019). *Shortage Of Radiologists In Australia*. Accessed: Dec. 7, 2020. [Online]. Available: https://blog.gorillajobs.com.au/shortage-of-radiologists-in-australia

[6] J. Dabass, S. Arora, R. Vig, and M. Hanmandlu, ''Segmentation techniques for breast cancer imaging modalities—A review,'' in *Proc. 9th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, Jan. 2019, pp. 658–663.

[7] J. Suckling, J. Parker, and D. Dance, ''The mammographic image analysis society digital mammogram database excerpta medica,'' in *Proc. Int. Congr. Ser.*, vol. 1069, 1994, pp. 375–378.

[8] S. R. Sternberg, ''Biomedical image processing,'' in *Prco. IEEE Comput.*, Jan. 1983, pp. 22–34.

[9] F. Shaukat, G. Raja, A. Gooya, and A. F. Frangi, ''Fully automatic detection of lung nodules in CT images using a hybrid feature set,'' *Med. Phys.*, vol. 44, no. 7, pp. 3615–3629, Jul. 2017.

[10] S. A. El-Regaily, M. A. M. Salem, M. H. A. Aziz, and M. I. Roushdy, ''Multi-view convolutional neural network for lung nodule false positive reduction,'' *Expert Syst. Appl.*, vol. 162, Dec. 2020, Art. no. 113017.

[11] N. M. Tosic, A. Samcovic, D. Nikolic, D. Drajic, and N. Lekic, ''An algorithm for detection of electromagnetic interference in high frequency radar range-Doppler images caused by LEDs,'' *IEEE Access*, vol. 7, pp. 84413–84419, 2019.

[12] T. M. A. Basile, A. Fanizzi, L. Losurdo, R. Bellotti, U. Bottigli, R. Dentamaro, V. Didonna, A. Fausto, R. Massafra, M. Moschetta, P. Tamborra, S. Tangaro, and D. La Forgia, ''Microcalcification detection in full-field digital mammograms: A fully automated computer-aided system,'' *Phys. Medica*, vol. 64, pp. 1–9, Aug. 2019.

[13] L.-K. Huang and M.-J.-J. Wang, ''Image thresholding by minimizing the measures of fuzziness,'' *Pattern Recognit.*, vol. 28, no. 1, pp. 41–51, Jan. 1995.

[14] L. A. Zadeh, K. S. Fu, and K. Tanaka Eds., *Fuzzy Sets and Their Applications to Cognitive and Decision Processes: Proceedings of the US–Japan Seminar on Fuzzy Sets and Their Applications, Held at the University of California, Berkeley, California*. New York, NY, USA: Academic, Jul. 2014.

[15] R. R. Yager, ''On the measure of fuzziness and negation part I: Membership in the unit interval,'' *Int. J. Gen. Syst.*, vol. 5, no. 4, pp. 221–229, Jan. 1979.

[16] S. Aja-Fernández, A. H. Curiale, and G. Vegas-Sánchez-Ferrero, ''A local fuzzy thresholding methodology for multiregion image segmentation,'' *Knowl.-Based Syst.*, vol. 83, pp. 1–12, Jul. 2015.

[17] P. K. Sran, S. Gupta, and S. Singh, ''Integrating saliency with fuzzy thresholding for brain tumor extraction in MR images,'' *J. Vis. Commun. Image Represent.*, vol. 74, Jan. 2021, Art. no. 102964.

[18] C. K. Lee and S. P. Wong, ''A mathematical morphological approach for segmenting heavily noise-corrupted images,'' *Pattern Recognit.*, vol. 29, no. 8, pp. 1347–1358, Aug. 1996.

[19] M. Hazarika and L. B. Mahanta, ''A new breast border extraction and contrast enhancement technique with digital mammogram images for improved detection of breast cancer,'' *Asian Pacific J. Cancer Prevention*, vol. 19, no. 8, p. 2141, 2018.

[20] D. A. Zebari, D. Q. Zeebaree, A. M. Abdulazeez, H. Haron, and H. N. A. Hamed, ''Improved threshold based and trainable fully automated segmentation for breast cancer boundary and pectoral muscle in mammogram images,'' *IEEE Access*, vol. 8, pp. 203097–203116, 2020.

[21] J. Canny, ''A computational approach to edge detection,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[22] A. Rampun, P. J. Morrow, B. W. Scotney, and J. Winder, ''Fully automated breast boundary and pectoral muscle segmentation in mammograms,'' *Artif. Intell. Med.*, vol. 79, pp. 28–41, Jun. 2017.

[23] P. Jaccard, ''The distribution of the flora in the alpine zone,'' *New Phytol.*, vol. 11, no. 2, pp. 37–50, 1912.

[24] T. Sørensen, ''A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons,'' *Biol. Skrifter*, vol. 5, no. 4, pp. 1–34, 1948.

[25] L. R. Dice, ''Measures of the amount of ecologic association between species,'' *Ecology*, vol. 26, no. 3, pp. 297–302, Jul. 1945.

[26] S. A. Taghanaki, Y. Liu, B. Miles, and G. Hamarneh, ''Geometry-based pectoral muscle segmentation from MLO mammogram views,'' *IEEE Trans. Biomed. Eng.*, vol. 64, no. 11, pp. 2662–2671, Nov. 2017.

[27] S. Biswas and D. Ghoshal, ''Blood cell detection using thresholding estimation based watershed transformation with Sobel filter in frequency domain,'' *Procedia Comput. Sci.*, vol. 89, pp. 651–657, Jan. 2016.

[28] J. R. Parker, *Algorithms for Image Processing and Computer Vision*. Hoboken, NJ, USA: Wiley, 2010.

[29] P. Kandhway, A. K. Bhandari, and A. Singh, ''A novel reformed histogram equalization based medical image contrast enhancement using krill herd optimization,'' *Biomed. Signal Process. Control*, vol. 56, Feb. 2020, Art. no. 101677.

[30] R. O. Duda and P. E. Hart, ''Use of the Hough transformation to detect lines and curves in pictures,'' *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.

[31] V. B. Bora, A. G. Kothari, and A. G. Keskar, ''Robust automatic pectoral muscle segmentation from mammograms using texture gradient and Euclidean distance regression,'' *J. Digit. Imag.*, vol. 29, no. 1, pp. 115–125, Feb. 2016.

[32] P. Shi, J. Zhong, A. Rampun, and H. Wang, ''A hierarchical pipeline for breast boundary segmentation and calcification detection in mammograms,'' *Comput. Biol. Med.*, vol. 96, pp. 178–188, May 2018.

[33] I. B. Y. Goodfellow, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1, no. 2. Cambridge, MA, USA: MIT Press, 2016.

[34] D. Karimi, H. Dou, S. K. Warfield, and A. Gholipour, ''Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis,'' *Med. Image Anal.*, vol. 65, Oct. 2020, Art. no. 101759.

[35] C. M. Bishop, ''Training with noise is equivalent to Tikhonov regularization,'' *Neural Comput.*, vol. 7, no. 1, pp. 108–116, Jan. 1995.

[36] A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser, K. Kurach, and J. Martens, ''Adding gradient noise improves learning for very deep networks,'' 2015, *arXiv:1511.06807*. [Online]. Available: http://arxiv.org/abs/1511.06807

[37] R. M. Zur, Y. Jiang, L. L. Pesce, and K. Drukker, ''Noise injection for training artificial neural networks: A comparison with weight decay and early stopping,'' *Med. Phys.*, vol. 36, no. 10, pp. 4810–4818, Sep. 2009, doi: 10.1118/1.3213517.

[38] S. M. Kwok, R. Chandrasekhar, Y. Attikiouzel, and M. T. Rickard, ''Automatic pectoral muscle segmentation on mediolateral oblique view mammograms,'' *IEEE Trans. Med. Imag.*, vol. 23, no. 9, pp. 1129–1140, Sep. 2004.

[39] R. J. Ferrari, A. F. Frère, R. M. Rangayyan, J. E. L. Desautels, and R. A. Borges, ''Identification of the breast boundary in mammograms using active contour models,'' *Med. Biol. Eng. Comput.*, vol. 42, no. 2, pp. 201–208, Mar. 2004.

[40] P. S. Vikhe and V. R. Thool, ''Detection and segmentation of pectoral muscle on MLO-view mammogram using enhancement filter,'' *J. Med. Syst.*, vol. 41, no. 12, p. 190, Dec. 2017.

[41] H. Soleimani and O. V. Michailovich, ''On segmentation of pectoral muscle in digital mammograms by means of deep learning,'' *IEEE Access*, vol. 8, pp. 204173–204182, 2020.

[42] R. H. Sherrier and G. A. Johnson, ''Regionally adaptive histogram equalization of the chest,'' *IEEE Trans. Med. Imag.*, vol. MI-6, no. 1, pp. 1–7, Mar. 1987.

[43] D. Tellez, G. Litjens, P. Bándi, W. Bulten, J.-M. Bokhorst, F. Ciompi, and J. van der Laak, ''Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology,'' *Med. Image Anal.*, vol. 58, Dec. 2019, Art. no. 101544.

[44] T. M. Lehmann, A. Kaser, and R. Repges, "A simple parametric equation for pseudocoloring grey scale images keeping their original brightness progression," *Image Vis. Comput.*, vol. 15, no. 3, pp. 251–257, Mar. 1997.
[45] *ImageJ Software Tool*. Accessed: Dec. 7, 2020. [Online]. Available: https://www.imagej.net/
[46] *National Institutes Of Health (NIH)*. Accessed: Dec. 7, 2020. [Online]. Available: https://www.nih.gov/
[47] Laboratory For Optical And Computational Instrumentation. The University Of Wisconsin-Madison Research Lab Of Dr. Kevin Eliceiri. Accessed: Dec. 7, 2020. [Online]. Available: https://www.eliceirilab.org/
[48] A. John, W. Huda, E. M. Scalzetti, K. M. Ogden, and M. L. Roskopf, "Performance of a single lookup table (LUT) for displaying chest CT images1," *Acad. Radiol.*, vol. 11, no. 6, pp. 609–616, Jun. 2004.
[49] D. R. Bull, *Communicating Pictures: A Course in Image and Video Coding*. New York, NY, USA: Academic, Jul. 2014, pp. 100–132.
[50] Z. Wang, M. Li, H. Wang, H. Jiang, Y. Yao, H. Zhang, and J. Xin, "Breast cancer detection using extreme learning machine based on feature fusion with CNN deep features," *IEEE Access*, vol. 7, pp. 105146–105158, 2019.
[51] L. M. Mina and N. A. M. Isa, "A fully automated breast separation for mammographic images," in *Proc. Int. Conf. BioSignal Anal., Process. Syst. (ICBAPS)*, May 2015, pp. 37–41.
[52] I. K. Maitra, S. Nag, and S. K. Bandyopadhyay, "Technique for preprocessing of digital mammogram," *Comput. Methods Programs Biomed.*, vol. 107, no. 2, pp. 175–188, Aug. 2012.
[53] U. Bick, M. L. Giger, R. A. Schmidt, R. M. Nishikawa, D. E. Wolverton, and K. Doi, "Automated segmentation of digitized mammograms," *Acad. Radiol.*, vol. 2, no. 1, pp. 1–9, Jan. 1995.
[54] I. C. Moreira, I. Amaral, I. Domingues, A. Cardoso, M. J. Cardoso, and J. S. Cardoso, "INbreast: Toward a full-field digital mammographic database," *Acad. Radiol.*, vol. 19, no. 2, pp. 236–248, 2012.
[55] J. E. E. de Oliveira, A. M. C. Machado, G. C. Chavez, A. P. B. Lopes, T. M. Deserno, and A. D. A. Araújo, "MammoSys: A content-based image retrieval system using breast density patterns," *Comput. Methods Programs Biomed.*, vol. 99, no. 3, pp. 289–297, Sep. 2010.
[56] F. Gao, T. Wu, J. Li, B. Zheng, L. Ruan, D. Shang, and B. Patel, "SD-CNN: A shallow-deep CNN for improved breast cancer diagnosis," *Comput. Med. Imag. Graph.*, vol. 70, pp. 53–62, Dec. 2018.
[57] D. Ribli, A. Horváth, Z. Unger, P. Pollner, and I. Csabai, "Detecting and classifying lesions in mammograms with deep learning," *Sci. Rep.*, vol. 8, no. 1, pp. 1–7, Dec. 2018.
[58] D. Saranyaraj, M. Manikandan, and S. Maheswari, "A deep convolutional neural network for the early detection of breast carcinoma with respect to hyper-parameter tuning," *Multimedia Tools Appl.*, vol. 79, nos. 15–16, pp. 11013–11038, Apr. 2020.
[59] T. C. Redman, *Data Driven: Profiting From Your Most Important Business Asset*. Brighton, MA, USA: Harvard Business Press, 2008.
[60] J. Arevalo, F. A. González, R. Ramos-Pollán, J. L. Oliveira, and M. A. G. Lopez, "Representation learning for mammography mass lesion classification with convolutional neural networks," *Comput. Methods Programs Biomed.*, vol. 127, pp. 248–257, Apr. 2016.
[61] A. Dubrovina, P. Kisilev, B. Ginsburg, S. Hashoul, and R. Kimmel, "Computational mammography using deep neural networks," *Comput. Methods Biomech. Biomed. Eng., Imag. Vis.*, vol. 6, no. 3, pp. 243–247, 2018.
[62] S. Rahimeto, T. G. Debelee, D. Yohannes, and F. Schwenker, "Automatic pectoral muscle removal in mammograms," *Evolving Syst.*, pp. 1–8, Nov. 2019, doi: 10.1007/s12530-019-09310-8.
[63] N. Tavakoli, M. Karimi, A. Norouzi, N. Karimi, S. Samavi, and S. M. R. Soroushmehr, "Detection of abnormalities in mammograms using deep features," *J. Ambient Intell. Humanized Comput.*, pp. 1–13, Dec. 2019, doi: 10.1007/s12652-019-01639-x.
[64] S. D. Tzikopoulos, M. E. Mavroforakis, H. V. Georgiou, N. Dimitropoulos, and S. Theodoridis, "A fully automated scheme for mammographic segmentation and classification based on breast density and asymmetry," *Comput. Methods Programs Biomed.*, vol. 102, no. 1, pp. 47–63, Apr. 2011.
[65] M. J. Ali, B. Raza, A. R. Shahid, F. Mahmood, M. A. Yousuf, A. H. Dar, and U. Iqbal, "Enhancing breast pectoral muscle segmentation performance by using skip connections in fully convolutional network," *Int. J. Imag. Syst. Technol.*, vol. 30, no. 4, pp. 1108–1118, Dec. 2020.
[66] A. Rampun, K. López-Linares, P. J. Morrow, B. W. Scotney, H. Wang, I. G. Ocaña, G. Maclair, R. Zwiggelaar, M. A. G. Ballester, and I. Macía, "Breast pectoral muscle segmentation in mammograms using a modified holistically-nested edge detection network," *Med. Image Anal.*, vol. 57, pp. 1–17, Oct. 2019.
[67] K. Vaidehi and T. S. Subashini, "Automatic identification and elimination of pectoral muscle in digital mammograms," *Int. J. Comput. Appl.*, vol. 75, no. 14, pp. 15–18, Aug. 2013.
[68] C. Chen, G. Liu, J. Wang, and G. Sudlow, "Shape-based automatic detection of pectoral muscle boundary in mammograms," *J. Med. Biol. Eng.*, vol. 35, no. 3, pp. 315–322, Jun. 2015.
[69] T Table (2021). *T Table | T Table*. Accessed: Jan. 22, 2021. [Online]. Available: https://www.tdistributiontable.com/

ABHIJITH REDDY BEERAVOLU is currently pursuing the M.S. degree in information systems and data science with Charles Darwin University, Casuarina, NT, Australia. His goal is to live free and come up with ideas that can help the people and the societies near me and around the world. He is also a Computer Science Enthusiast who is interested in anything related to computers. His research interests include reading books on History and making comparisons with the current world, to make sense of the reality and its progression. Mostly, he is also interested in reading and analyzing information related to cognitive and behavioral psychology and trying to integrate them into various technological ideas.

SAMI AZAM is currently a Leading Researcher and a Senior Lecturer with the College of Engineering and IT, Charles Darwin University, Casuarina, NT, Australia. He is also actively involved in the research fields relating to Computer Vision, Signal Processing, Artificial Intelligence, and Biomedical Engineering. He has several publications in peer-reviewed journals and international conference proceedings.

MIRJAM JONKMAN (Member, IEEE) is currently a Lecturer and a Researcher with the College of Engineering, IT, and Environment, Charles Darwin University, Casuarina, NT, Australia. Her research interests include biomedical engineering, signal processing, and the application of computer science to real-life problems.

BHARANIDHARAN SHANMUGAM (Member, IEEE) received the Ph.D. degree in cybersecurity. He is currently a Research-Intensive Lecturer with the College of Engineering, IT, and Environment, Charles Darwin University, Casuarina, NT, Australia. He has many publications in several top journals and conference proceedings. His main research interests includes around the field of cybersecurity, the IoT, and cyber risk management, and he spends his free time nurturing next-generation kids.

KRISHNAN KANNOORPATTI is currently a research-active Associate Professor with the College of Engineering, IT, and Environment, Charles Darwin University, Casuarina, NT, Australia. In addition to being a stellar academic and innovative researcher, he also has extensive experience working with government bodies in setting up data privacy policies at the national and state level.

ADNAN ANWAR (Member, IEEE) has worked as a Data Scientist with Flow Power, an energy management and solution company. He is currently a Lecturer and the Deputy Director of the post-graduate cybersecurity studies with the School of Information Technology, Deakin University. He has more than eight years of research and teaching experience in universities and research labs, including NICTA, La Trobe University, and The University of New South Wales. His research interests include the security research for critical infrastructures, including smart energy grid, SCADA systems, and application of machine learning and optimization techniques to solve cyber security issues for industrial and the IoT systems.

• • •