

B-CIR U-NET: A DEEP LEARNING APPROACH TO DETECT, CLASSIFY AND LOCALIZE BREAST CANCER

JISHNU ROYCHOWDHURY¹, TATHAGATA SARKAR², KAUSHIK SARKAR³, SURAJIT BARI⁴, PRANAB HAZRA⁵ and Dr. ANILESH DEY^{6*}

^{1,2}Bachelor's degree program in electronics and communication engineering in Narula Institute of Technology, India,

^{3,5}Associated with Narula Institute of Technology, Dept. of Electronics & Communication Engineering

⁴Assistant Professor, in the Department of Electronics and Communication Engineering of Narula Institute of Technology, Agarpara, Kolkata.

⁶Associate Professor of Electronics and Communication Engineering at Narula Institute of Technology, Agarpara, Kolkata.

* Corresponding Author

ABSTRACT

Breast cancer is one of the top causes of death of women globally; however it is highly treatable if detected at an early stage. The traditional approach for breast screening is x-ray mammography, which is known to be difficult for detecting cancer tumors early. The complex breast structure caused by the process of compression during imaging makes it harder to detect minor size disorders. Furthermore, inter- and intra-variation of breast tissues makes it challenging to attain high diagnostic accuracy using hand-crafted characteristics. In this paper, we propose a new method with the ease of deep learning, B-Cir U-Net inspired from the U-Net architecture, for the effective and early detection, localization and classification of breast cancer. The results show a high accuracy rate along with a high rate of sensitivity and specificity, proving that the proposed methodology might be effective in medical settings.

Index Terms— Augmentation, Breast Cancer, Convolution Neural Networks (CNN), Deep Learning, Image processing, Mammography, Separable Convolution, U-Net architecture

1. INTRODUCTION

Breast cancer is one of the most common but deadly disease, escalating worldwide among women, claiming lives of thousands each year. In 2020, about 2.3 million women were diagnosed with breast cancer making it world's most prevalent cancer. Standing on this era, it is difficult to avoid breast cancer, as it is affecting even the developed countries dreadfully according to the World Health Organization (WHO). To save lives, early detection of breast cancer is very necessary. To examine human breast for early detection of breast cancer, a process called mammography is done using low energy X-Ray (usually around 30 kVp). This simple screening protocol is accepted widely due to its cost effectiveness. Mammograms are inspected on a daily basis to search whether there are any abnormal lesions present or not. If, certain locations are found, the radiology experts detect the shape and type of such suspicious regions using mammograms. However, while observing small size abnormal structures (less than 1 mm), we face difficulty. Moreover, one might have trouble to distinguish between dense breast structures and that of lesions of similar contrast. Thus, the process of mammography might result in a high chance of occurrence of any error. Since cancer detection is a very

measured task, we can't afford any negligence in accuracy. So with the help of Medical Image Segmentation, we can detect, localize as well as extract detailed information such as nature and root cause of the suspicious regions of tumors and sent it for further diagnosis.

In deep learning, artificial intelligence moves to a level closer to human intelligence. In this paper, we have taken inspiration from the U-Net architecture, made it more advanced introducing Circular U-Net and tried to formulate an algorithm which not only detects and localizes the tumor's region with utmost precision but also classifies them as benign and malignant tumors. Development in automatic diagnosis system for breast cancer using mammography is a long time active track with achievements that apply different techniques to improve quality of diagnosis. In this work it has been tried to bring the accuracy nearer to 100% by employing the B-Cir U-Net which not only localizes the area of the cancer but also stacks them separately. Thus, an automated system can be an absolute remedy from the high numbers of mammograms and handle this process automatically.

2. RELATED WORK

Some related works like the "Breast cancer detection using convolutional neural networks for mammogram imaging system" *et.al* Y. J. Tan, K. S. Sim, F. F. Ting [1] in which they have designed a process of Breast Cancer Detection Using Convolutional Neural Networks (BCDCNN) aims to speed up the diagnosis process by supporting specialists in breast cancer diagnosis and classification in which their findings demonstrate that the suggested approach is more accurate than other current methods, the mass only and all arguments have been enhanced from 0.75 to 0.8585 and 0.608974 to 0.8271 accuracy, respectively. Another method of "Breast Cancer Detection Using Deep Learning", *et.al* Shaniar Tahir [2], described a deep learning strategy employing a variety of dataset models, achieving 97.35 percent accuracy and 0.98 AUC on the DDSM database, 95.50 percent accuracy and 0.97 AUC on the INbreast database, and 96.67 percent accuracy and 0.96 AUC on the BCDR database. A similar approach of "Breast Tumor Segmentation using U-NET", *et.al* Mirya Robin, Jisha John, Aswathy Ravikumar [3] in their research effort offered a UNET-based architecture for tumor area segmentation in histopathology images. The network is based on a fully convolutional network, and its architecture is being modified and enlarged in order to function with less training images and provide more accurate segmentations. With a small dataset, the suggested technique achieves an overall accuracy of 94.2. Based on the research advances stated above, we provide a novel network design inspired by U-net and modified to improve diagnosis findings, as explained in the next section.

3. MATERIAL AND METHODS

Because breast abnormalities vary in size, texture, and contrast, automatic identification of breast abnormalities from mammograms is a difficult undertaking. The dataset utilized in this study, as well as the network design, are addressed in this section.

3.1 Dataset Collection

The Curated Breast Imaging Subset of the Digital Database of Screening Mammography (CBIS-DDSM) [5], [8] is utilised in this work for both training and evaluating the created deep learning technique. This dataset comprises mammography images from 6,671 patients with normal, benign, and malignant cases verified by pathology thus comprising a total number of 10239 images. DICOM images with binary labels are supplied for lesion identification contains further information about the CBIS-DDSM dataset.

3.2 Data Preprocessing

The CBIS-DDSM dataset provides data for micro calcification and mass anomalies from 603 (152) and 692 (202) participants, respectively, for training (testing). Images from the left and right breasts with bilateral craniocaudal (CC) and mediolateral oblique (MLO) views are available for each patient. We consider the use of CC views only and consider the combination of MLO views as future work. As a pre-processing step, a set of Region-of-Interest (ROI's) represent the region with potential abnormality is selected from the original mammograms. The ROIs are selected to have the abnormality centered (as much as possible) within a 1024x1024 pixels. Histogram equalization process is used to normalize image contrast. Then we have applied augmentation to generate additional pseudo data with positional augmentation techniques like cropping, flipping, translation, rotation, and distortion technique by applying Gaussian noise as shown in fig 1. The repetition is done at different cropping factors. Moreover, the angle of rotation is different for a different rotation stage, i.e. there are 9 different angle of rotation. Same process is applied for Translation and the noise stage. Thus increasing 40 times the initial data samples, begetting a total of 4,09,560 data samples.

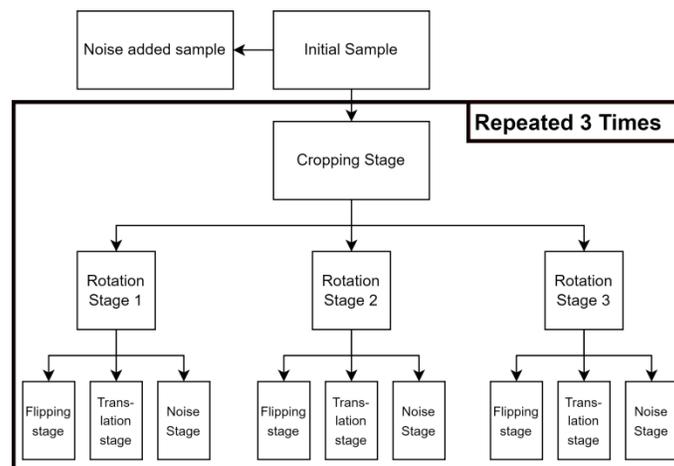


Fig 1. Illustration of the image augmentation technique

3.3 Data Leakage

Preventing data leaking is one of the methodology's most important objectives since, in the applicable data set, a single subject has many mammography Data (after augmentation). As a result, we can no longer use the 'train_test_split' command during splitting and must instead devise new logic to split the data at the individual category level. We accomplished this by manually allocating 70% of the data for training and the remaining 30% for testing. We could be certain that no data was leaking between the training and testing data sets. As a result, the samples utilized for testing are all unfamiliar to the network.

3.4 Proposed Method

The proposed method is inspired from the U-net architecture which is Circular U-Net. Now a question might arise why “circular” in the term? Well it's because the architecture seems to form a complete connection of a particular U-net block which looks like circular. The U-net is a well-known CNN that was designed for image segmentation specifically semantic segmentation. As an encoder-decoder architect, it is made up of consecutive convolution and deconvolution layers. U-net provides an intriguing architecture for image feature extraction, which is later applied in a variety of imaging applications [5]. Here, in our method, we have modified the U-net such that the skip connections are now having individual separable 2 dimensional convolutional block for further feature extraction in a particular step. Except for layers 9 to 19, which have two branches (up and down), the convolution process is implemented with a kernel size of (3x3) pixels, (1x1) padding, and (1x1) stride. Layers 9 to 19 employ a bigger convolution kernel with a size of (5x5) pixels, (2x2) padding, and (1x1) stride. The maximum pooling operation is set at (2x2) pixels. The encoding (or convolution) operation has been carried out by 2D CNN layer and the decoding (or deconvolution) operation has been carried out by 2D transposed CNN layer, and the skip connected convolutional block consists of 2D separable convolutional layers with a kernel size of (3x3), for the stochastic gradient descent, we have implemented “Adam” optimization algorithm [6]. We have incorporated a dropout of 0.1 along with batch normalization technique. Finally, in the last convolutional layer with a kernel size of (1x1), we are applying sigmoid activation function which will binarize the pixels for the localization and based on that we are finally classifying it as malignant or benign. The complete architecture, model summary and algorithm are shown in fig 2, table 1, and table 2 respectively.

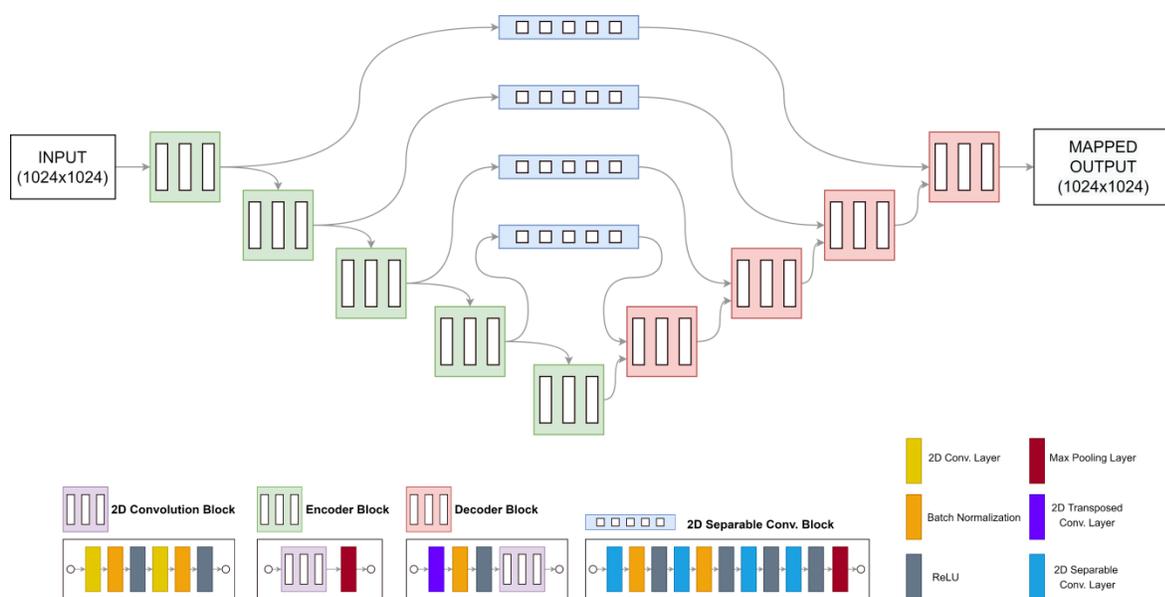


Fig 2. Basic illustration of the B-Cir U-net Architecture

**TABLE 1
 MODEL SUMMARY OF THE PROPOSED ARCHITECTURE**

Layer (type)	Output Shape	Param #	Connected to
img (InputLayer)	[(None, 1024, 1024,1)]	0	[]
conv2d_39 (Conv2D)	(None, 1024, 1024,16)	160	['img[0][0]']
batch_normalization_45 (BatchNormalization)	(None, 1024, 1024,16)	64	['conv2d_39[0][0]']
activation_37 (Activation)	(None, 1024, 1024,16)	0	['batch_normalization_45[0][0]']
max_pooling2d_17 (MaxPooling2D)	(None, 512, 512, 16)	0	['activation_37[0][0]']
dropout_16 (Dropout)	(None, 512, 512, 16)	0	['max_pooling2d_17[0][0]']
conv2d_41 (Conv2D)	(None, 512, 512, 32)	4640	['dropout_16[0][0]']
batch_normalization_49 (BatchNormalization)	(None, 512, 512, 32)	128	['conv2d_41[0][0]']
activation_39 (Activation)	(None, 512, 512, 32)	0	['batch_normalization_49[0][0]']
max_pooling2d_19 (MaxPooling2D)	(None, 256, 256, 32)	0	['activation_39[0][0]']
dropout_17 (Dropout)	(None, 256, 256, 32)	0	['max_pooling2d_19[0][0]']

conv2d_43 (Conv2D)	(None, 256, 256, 64)	256	18496	['dropout_17[0][0]']
batch_normalization_53 (BatchNormalization)	(None, 256, 256, 64)	256	256	['conv2d_43[0][0]']
activation_41 (Activation)	(None, 256, 256, 64)	256	0	['batch_normalization_53[0][0]']
max_pooling2d_21 (MaxPooling2D)	(None, 128, 128, 64)	128	0	['activation_41[0][0]']
dropout_18 (Dropout)	(None, 128, 128, 64)	128	0	['max_pooling2d_21[0][0]']
conv2d_45 (Conv2D)	(None, 128, 128, 128)	128	73856	['dropout_18[0][0]']
batch_normalization_57 (BatchNormalization)	(None, 128, 128, 128)	128	512	['conv2d_45[0][0]']
activation_43 (Activation)	(None, 128, 128, 128)	128	0	['batch_normalization_57[0][0]']
separable_conv2d_44 (SeparableConv2D)	(None, 128, 128, 128)	128	17536	['activation_43[0][0]']
batch_normalization_58 (BatchNormalization)	(None, 128, 128, 128)	128	512	['separable_conv2d_44[0][0]']
re_lu_33 (ReLU)	(None, 128, 128, 128)	128	0	['batch_normalization_58[0][0]']
separable_conv2d_45 (SeparableConv2D)	(None, 128, 128, 128)	128	17536	['re_lu_33[0][0]']
max_pooling2d_23 (MaxPooling2D)	(None, 64, 64, 128)	64	0	['activation_43[0][0]']
batch_normalization_59 (BatchNormalization)	(None, 128, 128, 128)	128	512	['separable_conv2d_45[0][0]']
dropout_19 (Dropout)	(None, 64, 64, 128)	64	0	['max_pooling2d_23[0][0]']
re_lu_34 (ReLU)	(None, 128, 128, 128)	128	0	['batch_normalization_59[0][0]']
conv2d_47 (Conv2D)	(None, 64, 64, 256)	64	295168	['dropout_19[0][0]']
separable_conv2d_46 (SeparableConv2D)	(None, 128, 128, 128)	128	17536	['re_lu_34[0][0]']
separable_conv2d_40 (SeparableConv2D)	(None, 256, 256, 64)	256	4672	['activation_41[0][0]']
batch_normalization_61 (BatchNormalization)	(None, 64, 64, 256)	64	1024	['conv2d_47[0][0]']
re_lu_35 (ReLU)	(None, 128, 128, 128)	128	0	['separable_conv2d_46[0][0]']
batch_normalization_54 (BatchNormalization)	(None, 256, 256, 64)	256	256	['separable_conv2d_40[0][0]']
activation_45 (Activation)	(None, 64, 64, 256)	64	0	['batch_normalization_61[0][0]']

separable_conv2d_47 (SeparableConv2D)	(None, 128, 128, 128)	17536	['re_lu_35[0][0]']
re_lu_30 (ReLU)	(None, 256, 256, 64)	0	['batch_normalization_54[0][0]']
conv2d_transpose_8 (Conv2DTranspose)	(None, 128, 128, 128)	295040	['activation_45[0][0]']
max_pooling2d_22 (MaxPooling2D)	(None, 128, 128, 128)	0	['separable_conv2d_47[0][0]']
separable_conv2d_41 (SeparableConv2D)	(None, 256, 256, 64)	4672	['re_lu_30[0][0]']
concatenate_8 (Concatenate)	(None, 128, 128, 256)	0	['conv2d_transpose_8[0][0]', 'max_pooling2d_22[0][0]']
batch_normalization_55 (BatchNormalization)	(None, 256, 256, 64)	256	['separable_conv2d_41[0][0]']
dropout_20 (Dropout)	(None, 128, 128, 256)	0	['concatenate_8[0][0]']
re_lu_31 (ReLU)	(None, 256, 256, 64)	0	['batch_normalization_55[0][0]']
conv2d_49 (Conv2D)	(None, 128, 128, 128)	295040	['dropout_20[0][0]']
separable_conv2d_42 (SeparableConv2D)	(None, 256, 256, 64)	4672	['re_lu_31[0][0]']
separable_conv2d_36 (SeparableConv2D)	(None, 512, 512, 32)	1312	['activation_39[0][0]']
batch_normalization_63 (BatchNormalization)	(None, 128, 128, 128)	512	['conv2d_49[0][0]']
re_lu_32 (ReLU)	(None, 256, 256, 64)	0	['separable_conv2d_42[0][0]']
batch_normalization_50 (BatchNormalization)	(None, 512, 512, 32)	128	['separable_conv2d_36[0][0]']
activation_47 (Activation)	(None, 128, 128, 128)	0	['batch_normalization_63[0][0]']
separable_conv2d_43 (SeparableConv2D)	(None, 256, 256, 64)	4672	['re_lu_32[0][0]']
re_lu_27 (ReLU)	(None, 512, 512, 32)	0	['batch_normalization_50[0][0]']
conv2d_transpose_9 (Conv2DTranspose)	(None, 256, 256, 64)	73792	['activation_47[0][0]']
max_pooling2d_20 (MaxPooling2D)	(None, 256, 256, 64)	0	['separable_conv2d_43[0][0]']
separable_conv2d_37 (SeparableConv2D)	(None, 512, 512, 32)	1312	['re_lu_27[0][0]']
concatenate_9 (Concatenate)	(None, 256, 256, 128)	0	['conv2d_transpose_9[0][0]',

			'max_pooling2d_20[0][0]'
batch_normalization_51 (BatchNormalization)	(None, 512, 512, 32)	128	['separable_conv2d_37[0][0]']
dropout_21 (Dropout)	(None, 256, 256, 128)	0	['concatenate_9[0][0]']
re_lu_28 (ReLU)	(None, 512, 512, 32)	0	['batch_normalization_51[0][0]']
conv2d_51 (Conv2D)	(None, 256, 256, 64)	73792	['dropout_21[0][0]']
separable_conv2d_38 (SeparableConv2D)	(None, 512, 512, 32)	1312	['re_lu_28[0][0]']
separable_conv2d_32 (SeparableConv2D)	(None, 1024, 1024, 16)	400	['activation_37[0][0]']
batch_normalization_65 (BatchNormalization)	(None, 256, 256, 64)	256	['conv2d_51[0][0]']
re_lu_29 (ReLU)	(None, 512, 512, 32)	0	['separable_conv2d_38[0][0]']
batch_normalization_46 (BatchNormalization)	(None, 1024, 1024, 16)	64	['separable_conv2d_32[0][0]']
activation_49 (Activation)	(None, 256, 256, 64)	0	['batch_normalization_65[0][0]']
separable_conv2d_39 (SeparableConv2D)	(None, 512, 512, 32)	1312	['re_lu_29[0][0]']
re_lu_24 (ReLU)	(None, 1024, 1024, 16)	0	['batch_normalization_46[0][0]']
conv2d_transpose_10 (Conv2DTranspose)	(None, 512, 512, 32)	18464	['activation_49[0][0]']
max_pooling2d_18 (MaxPooling2D)	(None, 512, 512, 32)	0	['separable_conv2d_39[0][0]']
separable_conv2d_33 (SeparableConv2D)	(None, 1024, 1024, 16)	400	['re_lu_24[0][0]']
concatenate_10 (Concatenate)	(None, 512, 512, 64)	0	['conv2d_transpose_10[0][0]', 'max_pooling2d_18[0][0]']
batch_normalization_47 (BatchNormalization)	(None, 1024, 1024, 16)	64	['separable_conv2d_33[0][0]']
dropout_22 (Dropout)	(None, 512, 512, 64)	0	['concatenate_10[0][0]']
re_lu_25 (ReLU)	(None, 1024, 1024, 16)	0	['batch_normalization_47[0][0]']
conv2d_53 (Conv2D)	(None, 512, 512, 32)	18464	['dropout_22[0][0]']
separable_conv2d_34 (SeparableConv2D)	(None, 1024, 1024, 16)	400	['re_lu_25[0][0]']
batch_normalization_67 (BatchNormalization)	(None, 512, 512, 32)	128	['conv2d_53[0][0]']

re_lu_26 (ReLU)	(None, 1024, 1024,16)	0	['separable_conv2d_34[0][0]']
activation_51 (Activation)	(None, 512, 512, 32)	0	['batch_normalization_67[0][0]']
separable_conv2d_35 (SeparableConv2D)	(None, 1024, 1024,16)	400	['re_lu_26[0][0]']
conv2d_transpose_11 (Conv2DTranspose)	(None, 1024, 1024,16)	4624	['activation_51[0][0]']
max_pooling2d_16 (MaxPooling2D)	(None, 1024, 1024,16)	0	['separable_conv2d_35[0][0]']
concatenate_11 (Concatenate)	(None, 1024, 1024,32)	0	['conv2d_transpose_11[0][0]', 'max_pooling2d_16[0][0]']
dropout_23 (Dropout)	(None, 1024, 1024,32)	0	['concatenate_11[0][0]']
conv2d_55 (Conv2D)	(None, 1024, 1024,16)	4624	['dropout_23[0][0]']
batch_normalization_69 (BatchNormalization)	(None, 1024, 1024,16)	64	['conv2d_55[0][0]']
activation_53 (Activation)	(None, 1024, 1024,16)	0	['batch_normalization_69[0][0]']
conv2d_56 (Conv2D)	(None, 1024, 1024,1)	17	['activation_53[0][0]']
Total params: 1,276,721			
Trainable params: 1,274,289			
Non-trainable params: 2,432			

TABLE 2
THE PROPOSED ALGORITHM

Algorithm 2: B-Cir U-Net Proposed Algorithm. The experiment in this paper used the default values $\alpha=0.0002$, $\beta_1=0.5$, $\beta_2=0.999$, $m=64$, $\epsilon=200$.

Data Pre-processing: S_i = Initial Mammography Data, C_{S_n} = Cropped image signal, R_n = Rotated image signal, F_n = Flipped image signal, T_n = Translated signal, D_{i^n} = distortion added image signal.

Image Processing for augmentation:

$C_{S_n}(x, y) \rightarrow S_i(f(\sigma) * (x, y))$; $f(\sigma)$ = cropping vector function; $1 < \sigma < n$

$R_n(x, y) \rightarrow C_{S_n}(f(\theta) * (x, y))$; $f(\theta)$ = trigonometric vector function; $0 < \theta \leq 180^\circ$

$F_n(x, y) \rightarrow R_n(\delta(x, y))$; $\delta = -1$ for horizontal flipping

$T_n(x, y) \rightarrow R_n(f(a, b) * (x, y))$; $f(a, b)$ = translation vector function of factors a, b

$D_{i^n}(x, y) \rightarrow G(x, y) + R_n(x, y)$; where $R_n(x, y)$ is the input signal, and $G(x, y)$ is a real white Gaussian noise function with single-sided noise power density (N_0)

Augment the processed data $[S_i(x, y), C_{S_n}(x, y), R_n(x, y), F_n(x, y), T_n(x, y), D_{i^n}(x, y)]$ altogether to form the final dataset.

Convert each sample data in the dataset into 1024 x1024 sizes.

Train the DNN with the augmented data and compute the initial weights.

Input for the DNN: Mammography Training set δ_1 , Testing set δ_2 .

$\alpha \rightarrow$ learning rate of the DNN $\beta_1, \beta_2 \rightarrow$ decay rates of average of gradients $m \rightarrow$ mini batch of data $\xi \rightarrow$ iteration step $\varepsilon \rightarrow$ the maximum number of iterations Output : ω^* , DNN weights
Train: for 1 to ε do for ξ steps do Sample minibatch of m examples $\{x^1, \dots, x^m\}$ from data training distribution $p_{data}(x)$ of δ_1 . Update the network ω^* by ascending its stochastic gradient: $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m [\log P(x^{(i)}) + \log(1 - P(x^{(i)}))]$ end end
The gradient based learning rule used in our algorithm is Adam optimization.

4. RESULTS AND DISCUSSIONS

The suggested architecture is designed in Python 3.7.6 [9] on an AMD(R) Ryzen™ 5 5600X workstation with 32 GB RAM and an NVidia GeForce GTX 1650 GPU is used to accelerate computations during training. The network is trained for a maximum of 200 iterations with a batch size of 64. If the loss value difference is smaller than 10^{-3} , the training can be stopped before reaching the maximum number of iterations. The ADAM optimizer was used as the default training optimizer algorithm [6], and images were randomly mixed before training. The network has been trained for the set of microcalcifications and masses. The training time, omitting the data pre-processing part, is less than 4 hours. An example of achieved results is shown in fig 4 and fig 5 along with the ground truth. We attained a training accuracy of 99.67% and a validation accuracy of 98.07% with very minimum losses as shown in fig 3. These findings surpass those published in [7] utilising well-known architectures such as AlexNet, VGGNet, and GoogleNet. One possible explanation for these findings is the use of two separate convolution kernels at the encoder tracks: the general 2D convolutional block and the separable 2D convolutional block. As a result, our suggested technique is a viable option for identifying complicated mammography patterns even when there are less data samples.

In a difficult diagnostic issue, the suggested B-Cir U-Net structure yields promising results. However, these are preliminary data that require more validation in order to precisely define the performance optimal settings. A more complicated network topology, for example, with higher kernel size, may help to enhance classification accuracy. Furthermore, the training set identification is still done manually by picking specific ROIs that incorporate the irregularities. Automatic ROI recognition or raw mammography processing is more practical and clinically beneficial.

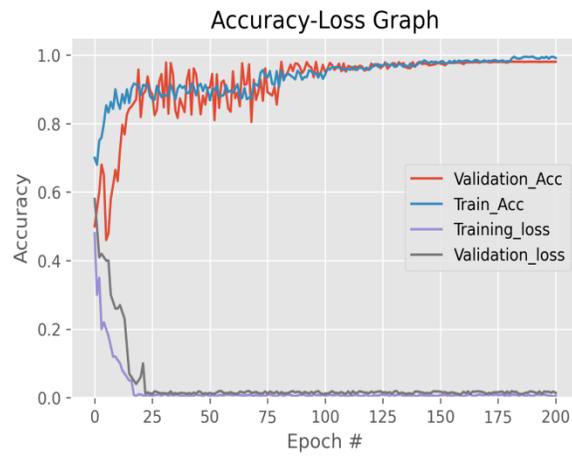


Fig 3. Accuracy Loss graph of the B-Cir U-net Architecture

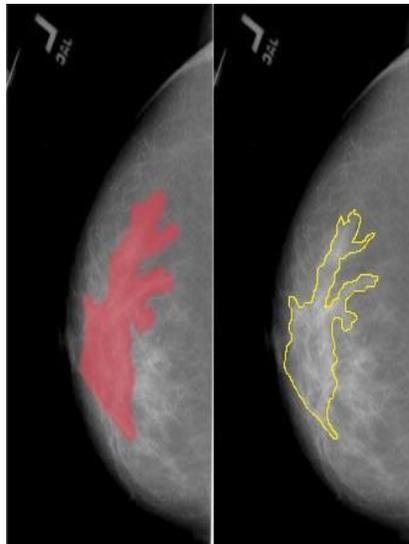


Fig 4. Microcalcification abnormality detection findings by B-Cir U-Net. The region labelled in red is the golden truth label, while the region labelled in yellow is the network identified region.

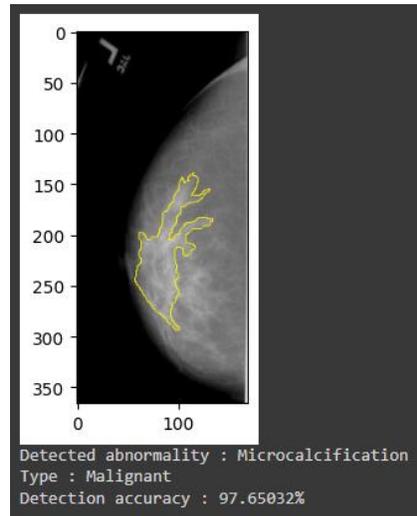


Fig 5. Output of our proposed method with the detection accuracy

5. CONCLUSION

We described a deep learning strategy for detecting breast cancer using mammograms in this paper. The suggested method is based on the development of convolution neural network, B-Cir U-Net, which demonstrates how resilient deep learning is in this application. As a future endeavor, numerous alternative variants of the suggested network design might be examined and evaluated. It is still uncertain whether network architecture is most suited to the texture of digital mammograms and how sensitive it is. The suggested technique may improve clinical identification of breast cancer, particularly in the early stages.

ACKNOWLEDGMENT

The authors wish to thank Dr. Anilesh Dey for the constant support and guidance in this research.

REFERENCES

1. Y. J. Tan, K. S. Sim and F. F. Ting, "Breast cancer detection using convolutional neural networks for mammogram imaging system," 2017 International Conference on Robotics, Automation and Sciences (ICORAS), 2017, pp. 1-5, doi: 10.1109/ICORAS.2017.8308076.
2. Tahir, Shaniar. (2019). Breast Cancer Detection Using Deep Learning. 10.13140/RG.2.2.16940.10889.
3. M. Robin, J. John and A. Ravikumar, "Breast Tumor Segmentation using U-NET," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 1164-1167, doi: 10.1109/ICCMC51019.2021.9418447.
4. <https://wiki.cancerimagingarchive.net/display/Public/CBIS-DDSM>
5. Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., Van Der Laak, J.A., Van Ginneken, B. and Sánchez, C.I., 2017. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42, pp 60- 88. DOI= <https://doi.org/10.1016/j.media.2017.07.005>
6. Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint

arXiv:1412.6980. <https://arxiv.org/abs/1412.6980>

7. Xi, P., Shu, C. and Goubran, R., 2018. Abnormality Detection in Mammography using Deep Convolutional Neural Networks. arXiv preprint arXiv:1803.01906 <https://arxiv.org/abs/1803.01906>
8. Rebecca Sawyer Lee, Francisco Gimenez, Assaf Hoogi, Kanae Kawai Miyake, Mia Gorovoy & Daniel L. Rubin. (2017) A curated mammography data set for use in computer-aided detection and diagnosis research. Scientific Data volume 4, Article number: 170177 DOI: <https://doi.org/10.1038/sdata.2017.177>
9. <https://www.python.org/>
10. Kooi, T., Litjens, G., Ginneken, B. Van, Gubern-mérida, A., Sánchez, C.I., Mann, R., Heeten, A. Den, Karssemeijer, N., 2017. Large scale deep learning for computer aided detection of mammographic lesions. Medical Image Analysis 35, 303– 312. DOI= <https://doi.org/10.1016/j.media.2016.07.007>
11. Ertosun, M.G. and Rubin, D.L., 2015, Probabilistic visual search for masses within mammography images using deep learning. In 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (pp. 1310-1315). IEEE. DOI= <https://dx.doi.org/10.1109/BIBM.2015.7359868>
12. Bekker, A.J., Greenspan, H. and Goldberger, J., 2016, April. A multi-view deep learning architecture for classification of breast microcalcifications. In Biomedical Imaging (ISBI), 2016 IEEE 13th International Symposium on (pp. 726-730). IEEE. DOI= <https://dx.doi.org/10.1109/ISBI.2016.7493369>