https://doi.org/10.33472/AFJBS.6.6.2024.1160-1172

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

Analysis of Breast Cancer Tumours with a Deep Neural Network Using MRI Images

[1]P. Bhargavi [2] T. Sarath [3]G V Ramesh Babu

[1]Assistant Professor, Department of Computer Science, Sri Padmavati Mahila Visvavidyalayam, Tirupati-517502

[2] Research Scholar, SCORE, VIT, Vellore

[3]Associate professor, Department of Computer Science, [3]Sri Venkateswara University Tirupati-517 502

Article Info

Volume 6, Issue 6, May 2024
Received: 03 March 2024
Accepted: 07 April 2024
Published: 14 May 2024
doi:10.33472/AFJBS.6.6.2024.1160-1172

Abstract

Breast cancer causes more deaths in women than any other type of cancer. Mammography is the primary screening test for breast cancer. Medical data from CT scans, PET scans, and MRIs are among the most widely used types of information. The use of data mining techniques has become essential for efficient and precise cancer prediction and detection since the work of analysing this massive amount of data has gotten increasingly difficult. Clinically relevant information can be mined from medical photographs to better aid in illness diagnosis and early detection, which is the primary focus of medical image mining. Patients need careful symptom observation and a prediction automatic system that can identify the tumour as benign or malignant in order to receive effective treatment. While its primary function as a generic convolutional neural network is to classify images where the input is an image and the output is a single label, in biomedical applications it also allows us to detect the presence of disease and pinpoint its exact location. This issue can be fixed using deep learning techniques. For the purpose of segmenting and prediction of tumour zone in mammography images, a Grad cam with soft-UNET based architecture is proposed. In order to determine the stage of breast cancer, the network relies on a fully convolutional network that has been upgraded and expanded in design to function with less training images and to provide more accurate segmentations with tumour height and width.

Keywords: Breast Cancer, Deep Learning Methods, UNET, GradCAM, Classification.

1. Introduction

When compared to other female-specific malignancies, breast cancer is the worst. Mammography is the primary diagnostic tool for detecting breast cancer at the screening stage. Breast cancer has been identified as the 25th leading cause of death worldwide [3]. About 50,000 women in India are diagnosed with this malignancy each year. Mammograms can detect three primary warning indicators indicative of cancer: masses, calcification, and architectural distortion. Therefore, it becomes even more crucial to discover and diagnose this malignancy as early as possible. In order to maximise the likelihood of a successful treatment outcome, early identification of breast cancer is crucial [2]. In high-risk females, breast MRI has a high detection rate for even the smallest of cancer tumours. With a sensitivity of 97%, breast tumours are more accurately diagnosed [1]. If the condition is diagnosed at an early stage, the pain and toxicity might be lessened. The disease can be detected in its early stages, long before any symptoms appear, by screening. The data mining methods of "clustering" and "classification" are particularly well-suited to the task of analysing breast cancer photos. The photos are clustered into different disease categories using clustering, an unsupervised classification method. It is crucial in medical diagnostics to select an effective clustering strategy that is tailored to the specific needs of the project at hand [4, 7]. Medical data from CT scans, PET scans, and MRIs are among the most widely used types of information. The use of data mining techniques has become essential for efficient and precise cancer prediction and detection since the work of analysing this massive amount of data has gotten increasingly difficult. The primary purpose of medical image mining is to aid in the diagnosis and early detection of disease by extracting clinically useful information from medical images [5, 6].

Patients need careful symptom observation and a prediction automatic system that can identify the tumour as benign or malignant in order to receive effective treatment. While its primary function as a generic convolutional neural network is to classify images where the input is an image and the output is a single label, in biomedical applications it also allows us to detect the presence of disease and pinpoint its exact location. This issue can be fixed using deep learning techniques.

For the purpose of segmenting the tumour zone in mammography images, this study proposes a GradCAM and UNET based architecture. In order to determine the stage of breast cancer, the network relies on a fully convolutional network that has been upgraded and expanded in design to function with less training images and to provide more accurate segmentations with tumour height and width.

2. Analysing Images

Segmenting an image into numerous regions by labelling each of its pixels is a common computer vision task. More information is provided than with object detection, which simply creates a box around the object, or picture categorization, which simply labels the object. Segmentation has many practical uses, including in medical imaging, clothing segmentation, flood map creation, autonomous vehicle navigation, and more.

Segmentation of images can be either:

- Label each pixel as part of a ***semantic segmentation***.
- Separate out ***individual instances*** of a given object by categorising their pixels.

3. Methodology

A study's methodology refers to the methods employed to gather data about the study's subject. The methodology section of a research article is where the reader is given the opportunity to assess the study's overall validity and dependability.

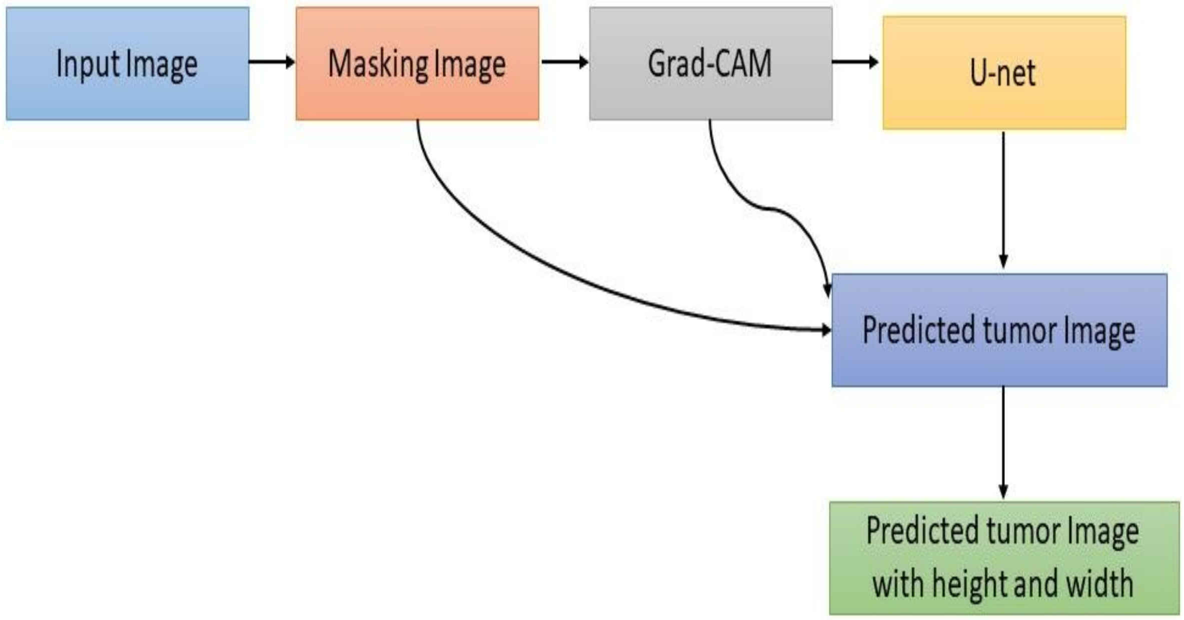


Figure 1: Steps in a Research Methodology

3.1. Masking

A digital mask can be used to obscure or show specific areas of an image. Masks are used to pick parts of an image for editing or to apply special effects in computer graphics. In an image editor, masks can be made from scratch or produced from pre-existing pictures. To make a stencilled effect, for instance, you could use a photograph as a mask. Graphic designers have many uses for masks, making them one of the most flexible tools at their disposal.

In computer graphics, masking is a method for disguising an object or area. Masking can be used to alter an image's appearance, remove distracting elements, or transform its form. Bit masking and alpha masking are two forms of masking. To hide pixels according to their colour values, bit masking can be used. When using alpha masking, pixels are hidden according to their opacity.

3.2. Grad-CAM

Zhou et al.'s CAM [10] approach is generalised in Gradient-weighted Class Activation Mapping (Grad-CAM) [11]. Since Grad-CAM doesn't require retraining existing models or modifying the original CNN architecture, it may be used to a wider variety of CNN architectures than CAM, which required that the CNN architecture not have completely connected layers. Grad-CAM can be used to selectively highlight one class in an image while omitting others, thanks to its class discriminatory nature. Unlike guided back-propagation and deconvolution, where class-specific visualisations are essentially similar, this approach allows for more flexibility between classes in the final product. Grad-CAM takes advantage of the gradient information streaming into the final convolutional layer, just before a fully connected layer (if present), which captures both high-level semantics and granular spatial information.

Grad-CAM works by first calculating the gradient of the class score, Y(c), with respect to the activations AK of the feature map in the final convolutional layer.

$$\frac{\partial Y^{(C)}}{\partial A_K(i,j)} \quad \text{--------(1)}$$

With the indices i and j, the gradients are pooled globally over the width (W) and the height (H) to get the neuron significance weights.

$$\omega_k^{(c)} = \frac{1}{H.W} \sum_{i=1}^{H} \sum_{j=1}^{W} \frac{\partial Y^{(C)}}{\partial A_K(i,j)} \quad \text{-------------- (2)}$$

After that, we derive the localization map by using:

$$L_{Grad-CAM}^{(c)}(x,y) = ReLU\left(\sum_k \omega_k^{(c)} A_K(x,y)\right) \quad \text{---------------- (3)}$$

where the activation of node k in the model's target layer at coordinates (x, y) is denoted by Ak (x, y). By using the RELU activation function, we can select and keep just the features that contribute positively to the target class.

3.3. U-Net

U-Net is a semantic segmentation method developed for diagnostic image analysis. The U-Net architecture is also utilised in several variations, including the Pix*Pix generator, making it one of the earliest deep learning segmentation models.

In 2015, Olaf Ronneberger [9] and his group created U-Net to use in their research with biological photos. Using less data and data augmentation to surpass the sliding window method, it won the ISBI competition.

When used to a training dataset, sliding window design excels at localization tasks. The method generates pixel-specific class labels by applying a local patch to each individual pixel. Two major issues with this design are, first, the large amount of overall redundancy that is developed as a result of overlapping patches. Second, it took a long time and a lot of resources to go through the training process. Due to these limitations, the design could not be used for many purposes. U-Net gets rid of both of these problems.

Previously, we discussed how segmentation is composed of categorization and localisation. To see why a U-Net is so well suited for segmentation, it is helpful to see how it accomplishes these two goals.

The architecture of U-Net inspired the network's moniker. The "U" shaped model consists of two networks and convolutional layers. The decoder comes after the encoder; hence the order is important. Both of these "what" and "where" concerns about segmentation are amenable to the U-Net.

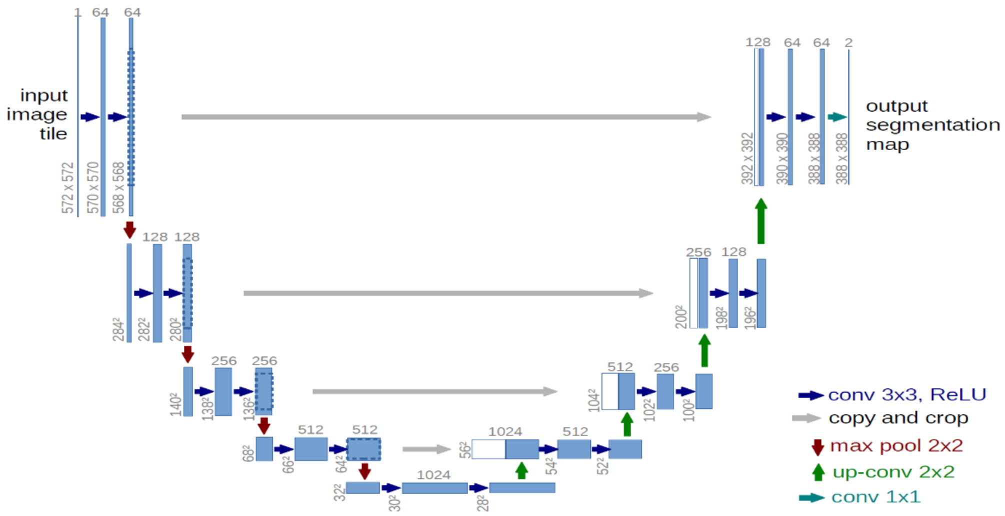
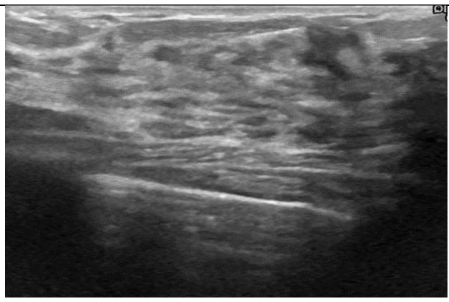
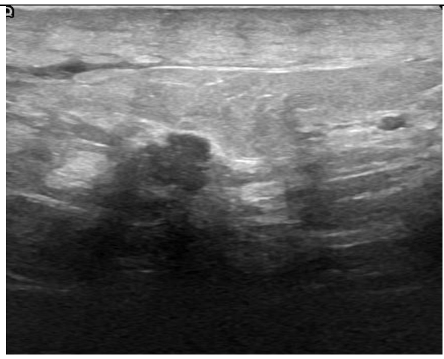


Figure 2: U-Net architecture [9]

4. Experimental Analysis

By using Kaggle mammograms of breast cancer images are considered for our analysis. First, within the python environment, the required packages are installed. The input photos, originally 500*500 pixels, are downsized to 256*256 so that we may examine them more closely. Figure 3 depicts the input image after making has been made to it to emphasise the diseased area, followed by the normal, malignant, and benign images.

Type of Image	Input image	Masking Image
Normal		
Malignant		

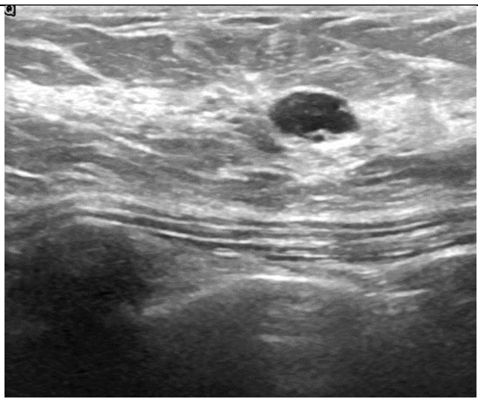
Benign		

Figure 3: Input Image and Masking image

Figures 4 and 5 depict the masking pictures used to hide the expected tumour.

The masking images of predicted tumor in the image are as shown in figure 4 and 5.

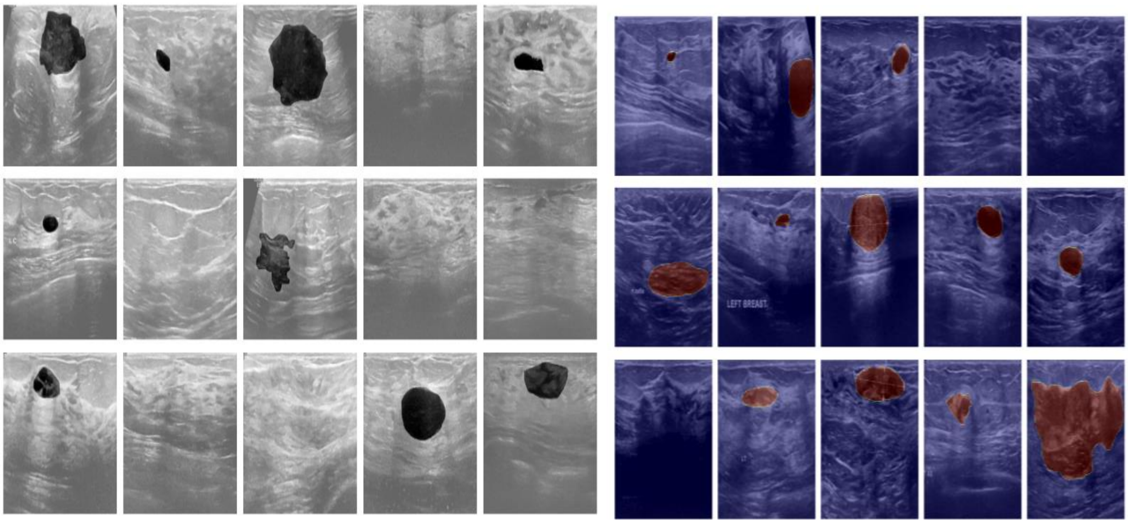


Figure 4: Making Image

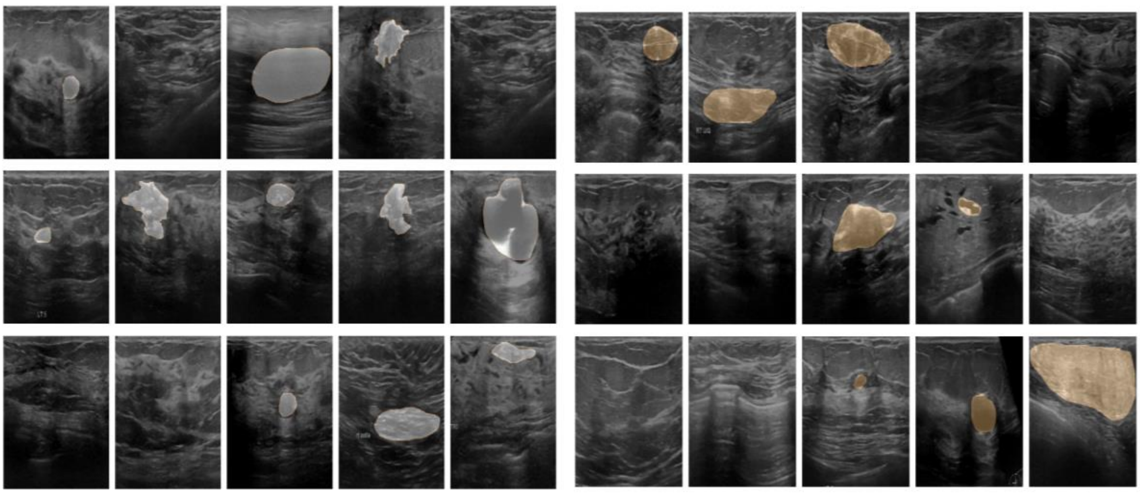


Figure 5: Making Image

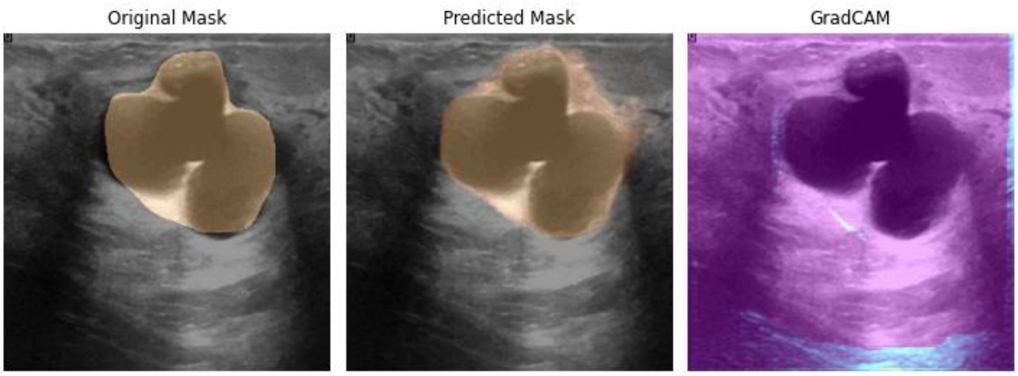
Then, the input images were processed using an Encoder Block, a Decoder Block, and an Attention Gate in order to densely locate the tumour. In order to accomplish this, we first use unpadded convolutions to lower the total dimension, and then we use Convolution blocks to determine which data is legitimate.
where the number of filters in the encoder design is increased while maintaining a steady reduction in image size by using max-pooling layers. Connecting earlier outputs to later levels in the decoder blocks leads to a reduction in the number of filters used in the convolutional layers.

For that purpose, a 64-by-default-filter double-layer convolution operation block was carried out after the primary operation had been carried out on the input parameters. Afterwards, a Batch Normalisation layer is added to the stack after these convolutional layers. Next, we'll construct the necessary encoding and decoding building components. From the top down, the encoder design will follow a pattern of sequential inputs. Our proposed encoder function will make use of a convolutional block consisting of two convolutional layers and a batch normalisation layer for each input layer. The decoder block's arguments will consist of the number of filters in use, the receiving inputs, and the input of the skip connection. Our model's Conv2DTranspose layers helped the up sample become input. Then, to get the final value of the skip connections, concatenate the input layer with the up-sampled layer. This output value is the result of a combined function and a convolutional block operation, as shown in Figure 6.

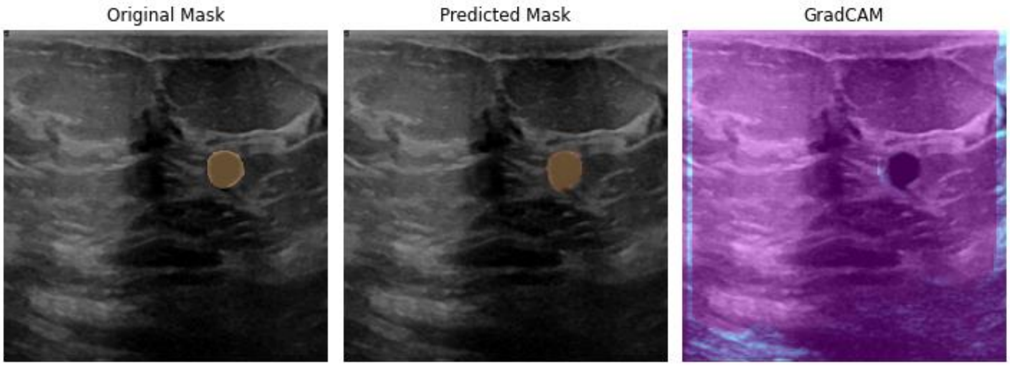
```
Model: "model_5"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_2 (InputLayer)            [(None, 256, 256, 3) 0
__________________________________________________________________________________________________
Encoder1 (EncoderBlock)         ((None, 128, 128, 32 10144       input_2[0][0]
__________________________________________________________________________________________________
Encoder2 (EncoderBlock)         ((None, 64, 64, 64), 55424       Encoder1[0][0]
__________________________________________________________________________________________________
Encoder3 (EncoderBlock)         ((None, 32, 32, 128) 221440      Encoder2[0][0]
__________________________________________________________________________________________________
Encoder4 (EncoderBlock)         ((None, 16, 16, 256) 885248      Encoder3[0][0]
__________________________________________________________________________________________________
Encoding (EncoderBlock)         (None, 16, 16, 512)  3539968     Encoder4[0][0]
__________________________________________________________________________________________________
Attention1 (AttentionGate)      (None, 32, 32, 256)  1771265     Encoding[0][0]
                                                                 Encoder4[0][1]
__________________________________________________________________________________________________
Decoder1 (DecoderBlock)         (None, 32, 32, 256)  2359808     Encoding[0][0]
                                                                 Attention1[0][0]
__________________________________________________________________________________________________
Attention2 (AttentionGate)      (None, 64, 64, 128)  443265      Decoder1[0][0]
                                                                 Encoder3[0][1]
__________________________________________________________________________________________________
Decoder2 (DecoderBlock)         (None, 64, 64, 128)  598080      Decoder1[0][0]
                                                                 Attention2[0][0]
__________________________________________________________________________________________________
Attention3 (AttentionGate)      (None, 128, 128, 64) 111841      Decoder2[0][0]
                                                                 Encoder2[0][1]
__________________________________________________________________________________________________
Decoder3 (DecoderBlock)         (None, 128, 128, 64) 147584      Decoder2[0][0]
                                                                 Attention3[0][0]
__________________________________________________________________________________________________
Attention4 (AttentionGate)      (None, 256, 256, 32) 27873       Decoder3[0][0]
                                                                 Encoder1[0][1]
__________________________________________________________________________________________________
Decoder4 (DecoderBlock)         (None, 256, 256, 32) 36928       Decoder3[0][0]
                                                                 Attention4[0][0]
__________________________________________________________________________________________________
conv2d_61 (Conv2D)              (None, 256, 256, 1)  33          Decoder4[0][0]
==================================================================================================
Total params: 10,200,101
Trainable params: 10,199,141
Non-trainable params: 960
__________________________________________________________________________________________________
```

Figure 6: convolutional block model

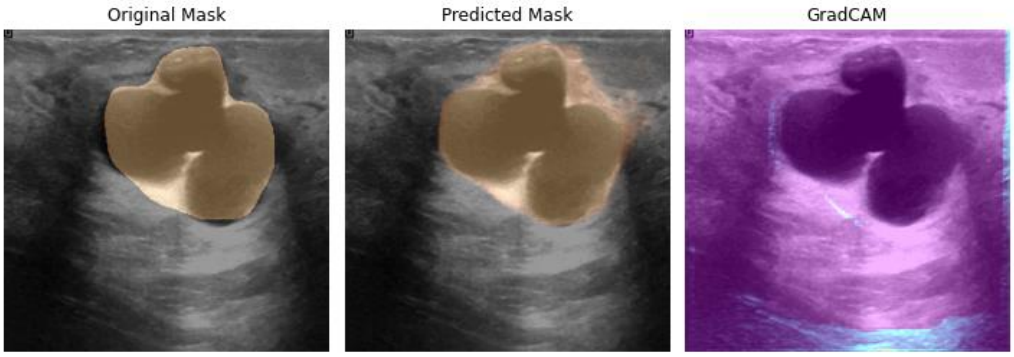
Then GradCAM is used to train the data and forecast the masking of the up sampling with varying accuracies over 30 iterations. Figure 7(a) displays the original mask, predicted mask, and GradCAM picture from a single input image time point. GradCAM's high-resolution error analysis and explanation capabilities make it ideal for tumour detection in input images. Figure 7 displays the limited analysis of only 5 time periods.



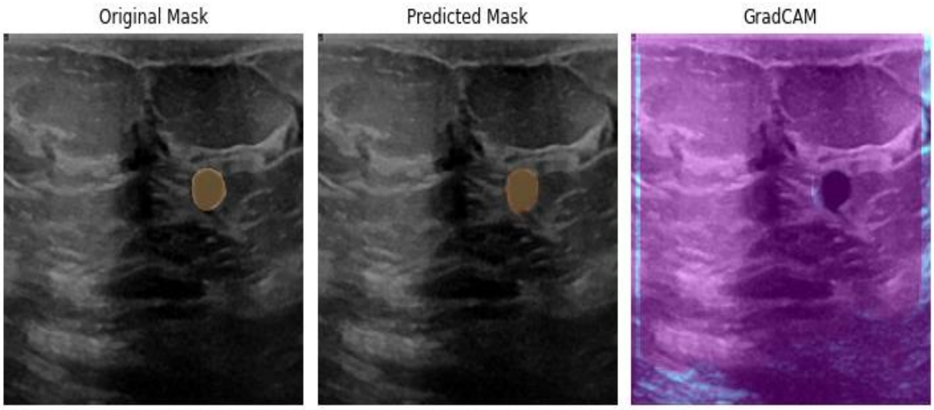
(a)



(b)



(c)



(d)

Figure 7: Making images using GradCAM

Next, visualized the trained and validated data's Model loss, accuracy, and model IOU. If the model has untapped potential beyond its current capabilities, as suggested by the visualisation results on Validation Data on IoU being significantly better than the results on Training Data, then this should be investigated further. Our approach appears to be promising for recognising the breast cancer tumour in input images, despite the fact that the loss is not ideal and actually rises at the end. Figure 8 depicts a model for visualising training data and IoU.

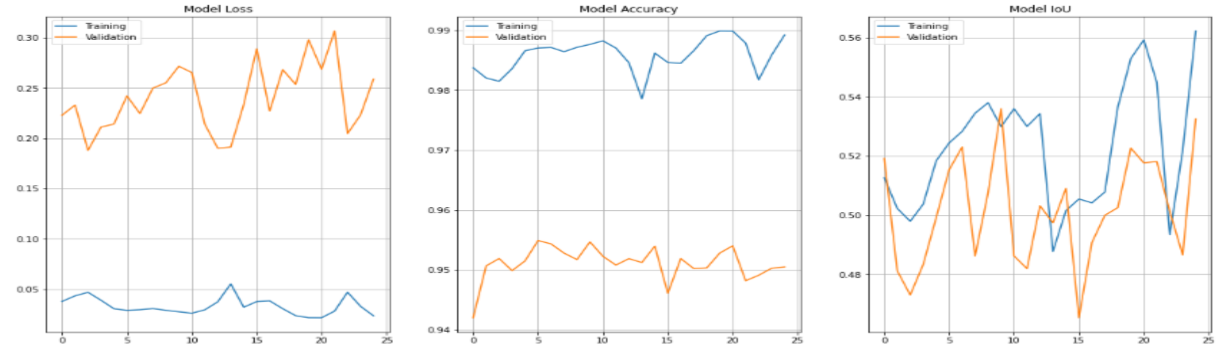


Figure 8: Model Loss, Accuracy, IoU

Because it operates on a convolutional network design, the tumour was located using the GradCAM masking picture and the U-Netand attention gates masks technique. As can be seen in Figure 9, the u net uses masking to improve segmentation accuracy in order to locate the cancerous region using less training data.

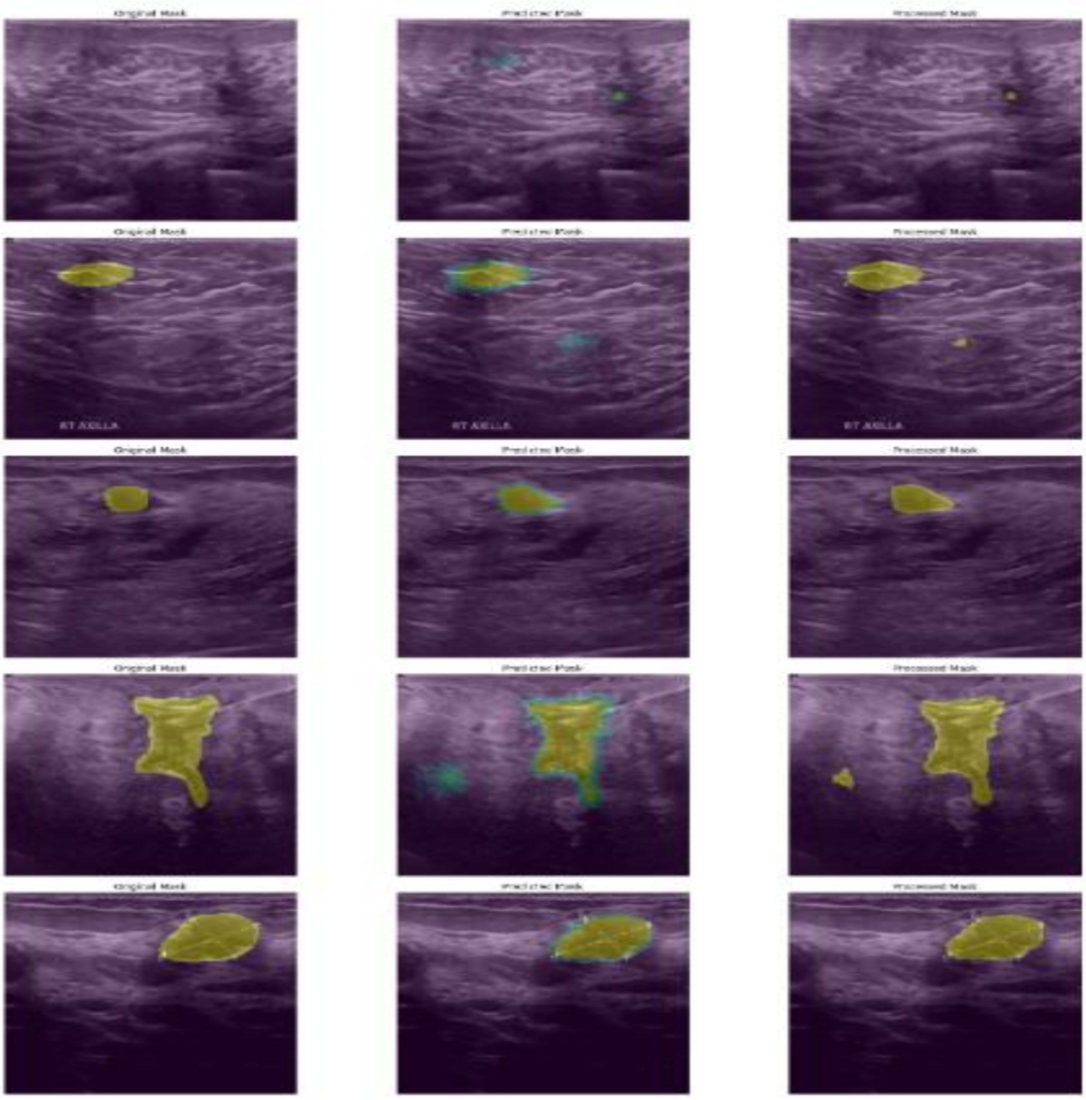


Figure 9: GradCAM and U-Net masking image of brest cance

Figure 10 depicts the results of using the U-Net architecture to create a precise segmentation map that used to determine the density of breast cancer.

[a]

[b]

[c]

[d]

[e]

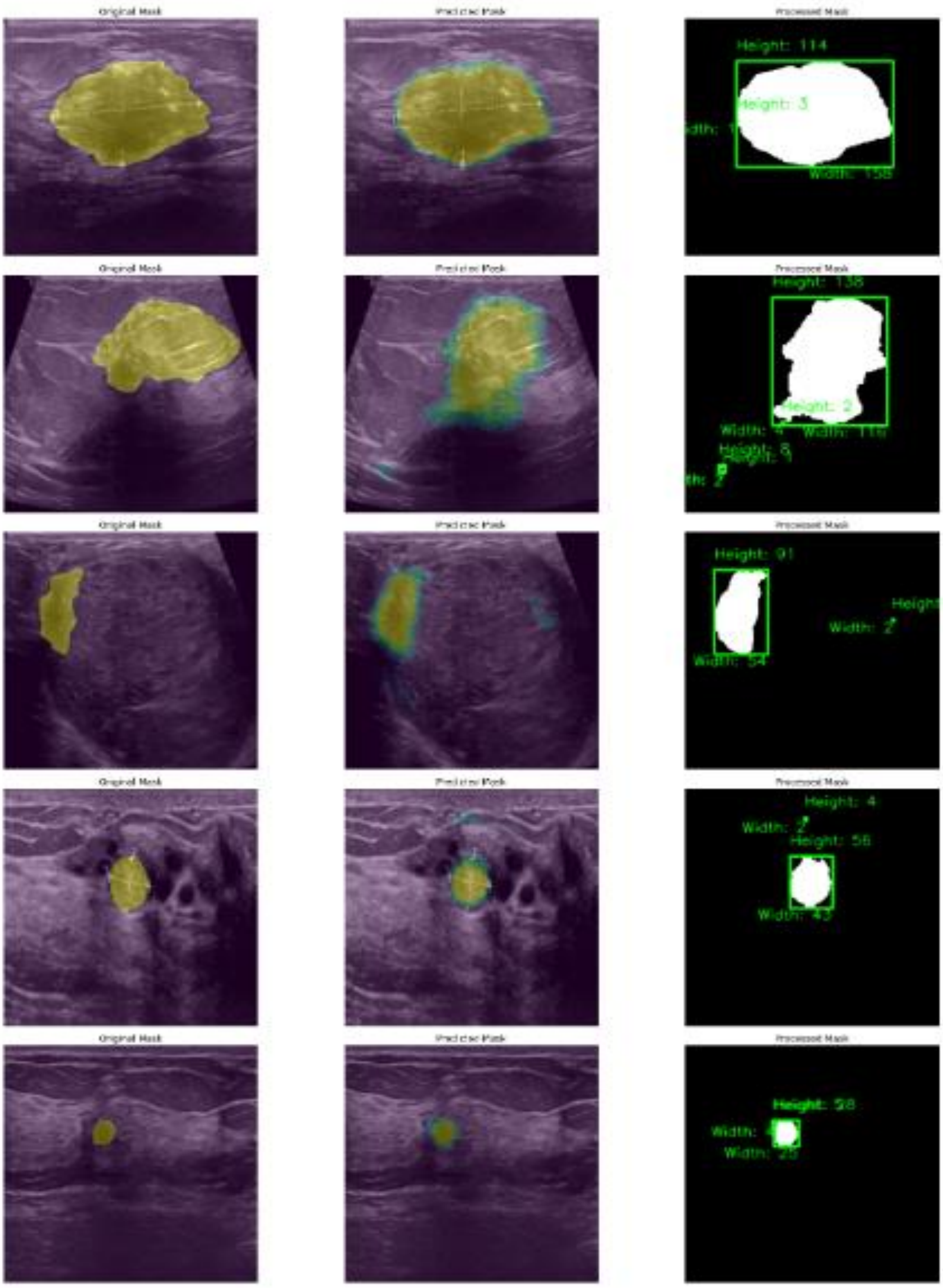


Figure 10: U-Net Architecture for Height and Width-Based Breast Cancer Detection.

Figure 10[a] shows a segmented tumour with a height of 114 and a width of 158, indicative of a very advanced stage of breast cancer. The height of the segmented tumour in figure 10 [b] is 138 and the breadth is 117, indicating a very advanced stage of breast cancer that is likely to spread. The segmented tumour in Figure 10 [c] has a height of 91 and a width of 54, indicating a moderate stage of breast cancer. The segmented tumour in Figure 10 [d] has a moderate stage of breast cancer based on its height (56) and width (43). Figure 10[e] shows a segmented tumour in a later stage of breast cancer, with a height of 16 and a width of 25.

In terms of the model giving high false positives and false negatives, we can observe and compare performance metrics such as precision, recall, and f1 score, all of which are at 98%. Because of the high rate of erroneous positive results, which mistakenly suggest the existence of a disease. While it's 98.7 percent accurate. It seems crazy good enough to represent the applied strategy for breast cancer classification.

5. Conclusion

Accurate early diagnosis is essential for breast cancer tumor predection. Images obtained from mammography are used in the diagnosis of breast cancer. Mammograms with a worrisome breast tumour are segmented with the help of a deep learning network. When used to tumour stage determination, the suggested GradCAM and U-net attention intensive approach for segmentation yields promising results. Height and width segmentation of the tumour can reveal useful information for determining the tumor's stage. GradCAM and U-net attention dense algorithm seems insanely good enough for classification of breast cancer as network is trained in more unique and diverse situation that increases the performance of the proposed algorithm with high false positive and false negative and accuracy of 98.7%.

References

[1] Shruthi Anand, Viji Vinod and anand rampure, (2015), “Application of Fuzzy c-means and Neural networks to categorize tumor affected breast MR Images” International Journal of Applied Engineering Research, ISSN 0973-4562 Vol. 10 No.64, pp: 274-28.

[2] Kowsalya, shanmuga priyaa, (2015), “A Survey on Diagnosis Methods of Breast Cancer using Mammography” International Journal of Engineering Technology Science and Research, ISSN 2394 – 3386 Vol. 02, Issue 4, pages,100-107.

[3] Delshi Howsalya Devi et al., (2016) “Outlier detection algorithm combined with decision tree classifier for early diagnosis of breast cancer” International Journal of Advanced Engineering Technology, E-ISSN 0976-3945, Vol. VII, Issue II, pp:93-98.

[4] S.Zulaikha Beevi , M.Mohammed Sathik , K.Senthamaraikannan, (2010) “A Robust Fuzzy Clustering Technique with Spatial Neighborhood Information for Effective Medical Image Segmentation” (IJCSIS) International Journal of Computer Science and Information Security, Vol. 7, No. 3.

[5] K.J. and G.W. Moore Cios, (2002), "Uniqueness of medical data mining," Artificial Intelligence in Medicine, pp. 1-24.

[6] R., Zhang, Y., Katta, (2002), "Medical Data Mining," Data Mining and Knowledge Discovery, pp. 305-308.

[7] N.S.Nithya, Dr.K.Duraiswamy, P.Gomathy, (2013), “A Survey on Clustering Techniques in Medical Diagnosis” International Journal of Computer Science Trends and Technology (IJCST) Vol. 1 Issue2.

[8] E. Barati, M. Saraee, A. Mohammadi, N. Adibi and M. R. Ahamadzadeh, (2011) "A Survey on Utilization of Data Mining Approaches for Dermatological (Skin) Diseases Prediction", Journal of Selected Areas in Health Informatics (JSHI): March Edition, page.1-11.

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox (2015), U-Net: Convolutional Networks for Biomedical Image Segmentation, computer vision and pattern recognition, Cornell University arXiv:1505.04597v1.

[10] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, (2016) "A. Learning deep features for discriminative localization", In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2921–2929.

[11] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D. (2017), "Grad-cam: Visual explanations from deep networks via gradient-based localization", In: Proceedings of the IEEE International Conference on Computer Vision, pp. 618–626.