CST4090 Individual Project

Analyzing the Impact of Synthetic Breast Ultrasound Image Data on an Existing App for Breast Cancer Detection

Dissertation

By,

Hajira Samrin

(M00899837)

Supervisor,

Chris Huyck

January 2024

This thesis is submitted in part of fulfilment of the requirements for the MSc. Data Science at Middlesex University, Jan 2023 to Jan 2024.

Acknowledgement

The journey of my master's has finally come to an end, and it has been the most enriching experience of my life. I am deeply grateful to everyone who has supported me throughout this journey. I thank God for blessing me with this wonderful opportunity and my beloved parents for being my steadfast support and helping me reach this milestone.

I would like to extend my immense gratitude to my supervisor, Professor Chris Huyck, for his guidance and support throughout my project. I am deeply thankful for his consistent reviews, productive meetings, valuable suggestions and his prompt responses to my emails.

I would like to thank the faculty members, Prof. Xiaohong Gao, Prof. Giovanni Quattrone, Prof. Clifford De Raffaele, Prof. Carlisle E George, Prof. David Windridge, Prof. Gill Whitney, Prof. Florian Kammueller, Prof. Oluwagbemi, Prof. Aniket Dixit, and Prof. Rebecca Mousa, for imparting their knowledge and helping me to understand the subjects.

I want to express my heartfelt thanks to my sisters, Nazia and Ayesha, for their guidance, support, and encouragement. My gratitude also extends to my nephew Fawad for his unwavering encouragement. I must thank my cat Kussi for simply being there, waiting for me while I write on my thesis and, unintentionally, teaching me the virtue of patience by walking across my laptop countless times. ☺

Lastly, I would like to thank the non-teaching staff, all the staff from Middlesex University's UniHelp and UniHub, and the Library for their assistance. My thanks also go out to all those who have supported me, both directly and indirectly, throughout this journey.

Abstract

Breast cancer is one of the most prevalent cancers in women globally. It is typically diagnosed through initial screenings, including physical examinations, mammography, and ultrasounds. In developing countries, particularly remote areas, access to mammography is often limited, making ultrasound a more common screening tool due to its wider availability. A significant challenge in these regions is the disproportionate ratio of the population to the number of radiologists and medical experts who can accurately diagnose the disease. To combat this shortage and facilitate the faster diagnosis of breast cancer with limited internet access, an application-based system has been developed. This system has achieved an accuracy of 87%, using a dataset containing 437 benign, 210 malignant, and 133 normal class images. However, one limitation of this system is the insufficient data for normal and malignant classes. A hypothesis was proposed to determine whether extending the dataset with synthetically generated data using a Generative Adversarial Network (GAN), which consists of a generator G to generate images using noise and a discriminator D that determines if the image resembles the training data, could improve system performance. The specific GAN used in this research was a Deep Convolutional GAN (DCGAN, in which G has convolutional-transpose layers and D has convolutional layers), which generated images for the malignant and normal classes to equalize the dataset, providing 437 images for each class. Initially, the existing system was replicated with an accuracy of 86%. This system was then trained using 10-fold cross-validation (System1), which achieved an accuracy of 93.432%, indicating a significant improvement. Although System 1 exhibited a slight reduction in sensitivity for the malignant class, when new data from DCGAN was added for training (System 2), there was an approximate 1.5% increase in accuracy (94.929%) and enhancements in sensitivity and specificity across all classes. These results support the hypothesis, suggesting a positive impact of synthetic data augmentation in model training.

Table of Contents

1. Introduction

Breast Cancer is the most prevalent cancer in women (*WCRF International, 2022*). In 2020, breast cancer was recorded as the most diagnosed cancer in women, especially in developing countries. In 2020, it was estimated that 685,000 deaths occurred due to breast cancer worldwide (*Arnold, M. et al., 2022*).

Breast cancer can be diagnosed through breast screening, which includes physical examination, mammography, ultrasound and MRI. As mentioned by *Bevers, T.B. et al., (2009)*, Breast ultrasonography may be a valuable screening complement to mammography in evaluating high-risk women with thick breasts.

In countries like India, mammography is not widely available (*Mehrotra, R et al., 2022*). Furthermore, in research carried out by *Devolli-Disha, E. et al., (2009)*, it was observed that ultrasound gave better sensitivity and specificity than mammography in patients exhibiting symptoms of breast cancer, which in turn was better for breast cancer detection. Having an app with increased accuracy and sensitivity in breast cancer detection will help reduce the issue of low radiologists-to-population ratio, especially in developing countries such as India (*Mehrotra, R et al., 2022*), thereby helping in faster detection of cancer and saving lives.

The existing system deals with breast cancer detection using an Android app. The accuracy of this system is 87.50% with varied sensitivity towards the 3 different classes of breast cancer (Malignant 0.95, Benign 0.86, normal 0.56). Here we can observe that sensitivity is lower for benign and much lower for normal when compared to that of malignant type. This means that the false negatives are high for benign and very high for normal type. One cause for such varied sensitivities is the lack of data in the normal class and the second is the imbalance of data between the benign and malignant classes (*Addala, V., 2023)*. This system comprises of MobileNetV2 model, which is one of the most efficient lightweight models, making it the best choice for an application-based system (*Sandler, M. et al. 2018*). Hence in this research, the existing system by Addala, V., (2023) is selected as the baseline model for the classification task.

Medical images, in general, are hard to obtain due to privacy concerns. Medical data collection is a time-consuming and costly process that necessitates the participation of researchers and radiologists. The dataset by *Al-Dhabyani et al. (2020)*, used in this research, has been cited 781 times. Other research on breast cancer ultrasound often relies on private datasets, which are typically accessible either through payment or via extensive privacy-related procedures. The study by *Thomas et al. (2023)* compiled ultrasound images from four different datasets, amassing a total of 1,154 images. This compilation includes 637 images from *Al-Dhabyani et al. (2020)* (the dataset used in this research), which constitutes the largest segment of this new dataset and still has data imbalance issues. For these reasons, researchers have come up with data augmentation techniques such as image rotation, image flipping and image scaling. However, the variations using these techniques are relatively low, thus paving the need for

image generation using deep learning techniques (*Frid-Adar, M. et al. 2018*). Since Breast Cancer ultrasound data is not largely available due to the aforementioned concerns, in this work, the focus is on solving the imbalanced data issue by introducing Synthetic Image generation using deep-learning techniques and checking the system for increased accuracy and sensitivity for all classes. (*Sedigh, P et al., 2019)*

AI in the medical field has helped in a faster, more accurate and more convenient diagnosis of disease. (*Davenport, T. et al., 2019*). Due to the widespread availability of ultrasound and its affordability (*Dan, Q. et al., 2023*) when compared to mammography in developing countries, using an automated system to evaluate the ultrasound images of breasts, taken at a stage of routine check-ups or when the patient exhibits symptoms, can help medical professionals track and evaluate the type of cancer at an early stage.

2. Brief Background

2.1. Existing systems:

The existing systems use Ultrasound images of breasts to detect and classify the types of breast cancer. The Author, *Addala, V. 2023*, has used different deep-learning techniques like CNN MobileNet V2 as the base deep-learning model. Currently, most of the research work deals with the dataset of Breast Mammography. Here the focus is on Ultrasound images of breasts as it was considered for population from all economic backgrounds. In the article by *Eroğlu, Y., Yildirim, M. and Çinar, A. (2021),* a hybrid model with three CNN models (Alexnet, Resnet50 and MobileNet) was built for feature extraction and this was then sent in for classification to the SVM and KNN classifiers.

The solution proposed by *Addala, V., (2023)* involves MobileNet-V2, a type of Convolution Neural Network (CNN) to learn from the dataset, detect breast cancer and classify the image as one of the three classes (Malignant, Benign and Normal). The author then used Tensor Flow Lite to convert this model to a mobile-friendly model, to integrate it within an Android application. The current model gives an accuracy of 87.50% and different sensitivities as mentioned in section 1. The Author has also focused on the explainability of code. The data was train

Eroğlu, Y., Yildirim, M. and Çinar, A. (2021) have proposed a solution that has a hybrid model for feature selection. Using Hybrid architecture of CNN with three models, Alexnet, Resnet50 and MobileNet, the authors have developed a model to extract the best features which are then fed to mRMR (Minimum Redundancy Maximum Relevance) which selected the best features produced by the three models and these best features are then sent to classifiers (SVM (Supposrt Vector Machine) and KNN (K Nearest Neighbours)) for classification of the breast ultrasound image. The system had the highest accuracy of 95.6% by the SVM classifier. The data was split into 80% for training and 20% for testing.

Ragab, M., Albukhari, et al. (2022) proposed a novel system for data extraction called the 'Ensemble Deep-Learning-Enabled Clinical Decision Support System for Breast Cancer Diagnosis and Classification (EDLCDS-BCDC).' This system integrates VGG-16, VGG-19,

and SqueezeNet for feature extraction and employs Cat Swarm Optimization for the classification of breast cancer ultrasound images. The process within this system starts with segmentation to distinguish between tumours and non-tumours, followed by feature extraction. Subsequently, these features are used to classify the ultrasound images. The system achieved a peak accuracy of 97.09%. The type of validation method used is not explicitly mentioned in this article.

One of the issues with the existing systems is the imbalanced and small dataset. The current solution by *Addala, V., (2023)*, by *Eroğlu, Y., Yildirim, M. and Çinar, A. (2021)* and by *Ragab, M., Albukhari, et.al. (2022)* is based completely on the existing dataset. The dataset used is imbalanced and does not contain enough training data for the classes. Hence the system by *Addala, V., (2023)* has the lowest sensitivity to the Normal class than any other class as Normal has the lowest number of Ultrasound images. This issue has affected both the accuracy and sensitivity of the system. Using data augmentation to expand the dataset, the system can be tested to check if there is any room for improvement with more data to train and test.

The system by *Eroğlu, Y., Yildirim, M. and Çinar, A. (2021)* and *Ragab, M., Albukhari, et.al. (2022)* includes hybrid models and adds computational burden on the system. Since the goal of this research is to work on a system, which can work in any smartphone to help identify the disease faster on a mobile-based application, this hybrid model is not suitable. MobilenetV2 is a lightweight CNN model as explained by *Sandler, M. et al.* (*2018*) and hence, it is the perfect choice for this research.

2.2. Understanding the selected model:

Convolutional Neural Networks (CNNs) are considered to be the best-suited choice for computer vision tasks. Early CNNs such as AlexNet were complex and computationally expensive. Hence the need to find CNNs with less computational cost, good accuracy and lightweight size, especially for use in small or resource-limited devices led to the creation of a new CNN called the MobileNets by *Howard, A.G., et al, 2017*. The MobileNets were faster and more computationally efficient than the other networks due to the type of convolution called "Depth-wise Separable convolution" which is a combination of depth-wise convolution and point-wise convolution.

What is a convolution?

CNN is a type of Artificial Neural Network(ANN) and is named after the linear mathematical operation of matrices named convolution. In a traditional fully connected neural network (often just called a neural network), every neuron in one layer is connected to every neuron in the next layer. This means that if you have a fully connected layer with **m** neurons and it's connected to another fully connected layer with **n** neurons, there will be **m * n** connections between them. Each of these connections has its associated weight, and the number of parameters can become very large, especially with high-dimensional input data, like images. To increase efficiency, it was good to just extract features from the local area rather than the entire image, such that the next neuron in the next layer receives only the selected features from the local area, thereby reducing the connections. For example, if the area selected is **m/6** then we have only **m/6 * n**

connections, assuming that the weights are kept fixed for the neurons of the next layer (i.e., weights of **m/6** same as **n**), causing an effect of sliding a filter. This helps in identifying the important features of an image from all the places.(*Albawi, S., et.al. 2017*)

The standard convolution process is shown in Figure 1. The computational cost of a standard convolution is $D_K^2*D_F^2*M*N$, as proved by *Bendersky2, E. (2003)*, where D_K is kernel size, D_F is the input feature map size, M is input channels, and N is output channel filters.

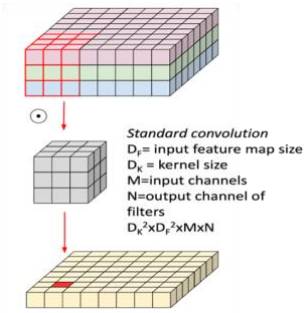


Figure 1: Standard convolution process, that uses 3x3x3 kernel over all channels of input feature, source *Bendersky2, E. (2003)*

What is Depthwise Separable convolution?

The reason MobileNets are faster and more efficient than other lightweight models is because of the depthwise separable convolution (*Howard, A.G., et al, 2017*). This convolution is a combination of depth-wise convolution and point-wise convolution. This is depicted in Figure 2.

Figure 2a represents the working of depth-wise convolution, which is a type of convolution similar to the standard convolution, but the difference is that convolution operation is performed separately on each channel hence having the number of kernels same as that of the number of channels. Unlike standard convolution in which the convolution operation sums up across all channels, in depthwise convolution the mapping is done separately, and the output of this convolution is stacked layers of mapped data for each channel(*Howard, A.G., et al, 2017*) (*Chollet, F., 2017*). Here since the kernel is separate, the output channel for each kernel is 1. So, the computational complexity is $D_K^2*D_F^2*M*1$, which is the same as 8 to $D_K^2*D_F^2*M$ (*Bendersky2, E. 2003).*

Figure 2b depicts the pointwise convolution. Pointwise convolution is also called a 1x1 convolution as the kernel size in this convolution is 1x1. This convolution is performed across all the channel layers, sums up and results in a single-layer output (*Howard, A.G., et al, 2017*) (*Chollet, F., 2017*). So, the computational complexity is $1^2*D_F^2*M*N$, which is the same as D_F^2*M*N (*Bendersky2, E. 2003).*

The sum of the depthwise convolution and pointwise convolution is Depthwise Separable convolution with a computational complexity of $D_K^2 * D_F^2 * M + D_F^2 * M * N$, making it 8 to 9 times faster than the standard convolution (*He, Y., et.al. 2019*).

Figure 2 shows a 2D image with RGB channels. In Figure 2a, we can see the separate kernels performing the convolution operation and in Figure 2b the output from the depthwise convolution then undergoes pointwise convolution, and both together form depthwise separable convolution.

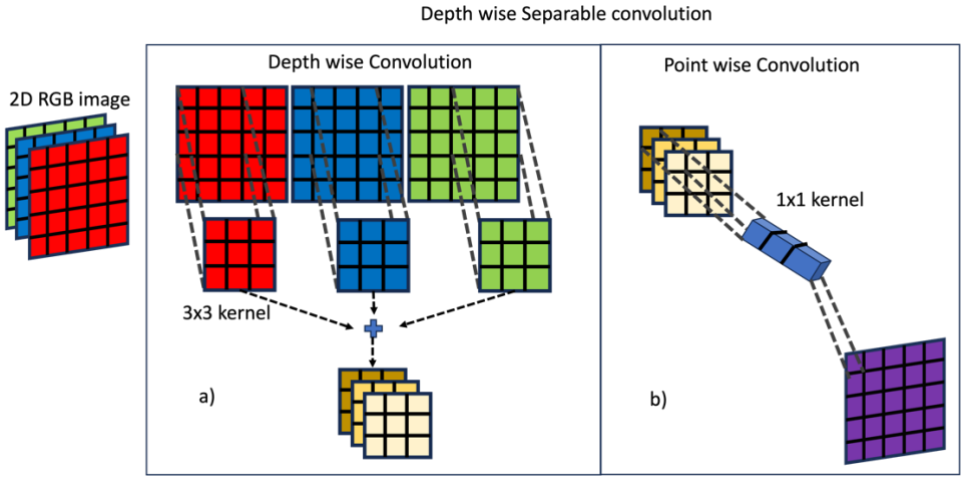


Figure 2: The figure depicts the working of a depth-wise separable convolution, **a)** Depth-wise convolution process, here 3x3x1 kernel performs convolution for each channel and the result is not summed up but retained as separate channels, **b)** Point-wise convolution process which uses 1x1x3 kernel, that sums convolution on a single cell across all channels and sums up the result as 1 layer.

What is MobileNetV2?

MobileNets were designed to operate efficiently on devices with limited computational power and to optimize latency in conjunction with a compact network structure. It overcomes constraints in both size and speed. Distinguished by its use of Depth-wise Separable convolutions, MobileNet stands out among compact network models due to its balance of speed and size. Apart from the first layer, Depth-wise Separable convolutions are employed throughout the network. Furthermore, all layers, aside from the final one (which will be discussed later in this research), implement batch normalization and ReLU (Rectified Linear Unit). Considering depth-wise and point-wise as distinct layers, MobileNet comprises 28 layers. It has been proven to be the most efficient model in terms of computational complexity and performance when compared to alternatives like VGG16, GoogleNet, AlexNet, and SqueezeNet (*Howard, A.G., et.al. 2017*)

MobileNetV2, a variation of MobileNetV1 (formerly referred to as just MobileNet before MobilNetV2 was found), retains the original structure of MobileNetV1 while increasing the system to achieve better accuracy, making it an enhanced and better version of MobileNetV1. One of the major updates in MobileNetV2 is the linear bottleneck layer. ReLU activation maintains a linear transformation of the input and introduces non-linearity so that the model can learn complex features of the data. The input transformations for each channel are linear

and the information is connected from one channel to another. If a channel is collapsed, then the information of that collapsed channel can be preserved due to that connection, but this is achievable only in the dimensional subspace of the input space which is relatively lower. This information is captured by the linear bottleneck layer, assuming that the input subspace is of low dimension. (*Sandler, M. et al. 2018).* Average pooling in this architecture reduces the spatial dimension of input data to 1, which is then sent to the last convolution layer. (*Howard, A.G., et.al. 2017*)

(a)

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

(b)

Input	Operator	Output
$h \times w \times k$	1x1 conv2d , ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

Figure 3: **a)** t is the expansion factor and applies only to the bottleneck layers, n is a sequence of 1 or more layers repeated n times, s is the stride for every first layer of a sequence and all other layers use stride 1, the kernel size is fixed to a 3x3, and all the layers use Depth wise Separable convolution **b)** this table shows the transformations, with an expansion factor t, and the transformation of k to k ′ and stride s

The network architecture for a mobilenetV2 model is shown in Figure 3(a). The first layer is a convolution layer (32 filters), and the next 19 layers are residual bottleneck layers. The activation function used in the model is ReLU6, which is another variation of ReLU that transforms input values between 0 and 6. Batch normalization and dropout layers are also used in the model. The kernel size is 3x3 and it is a fixed size throughout the network. Figure 3(b) gives the architecture of the residual bottleneck layer. These extra layers help capture the information and solve the vanishing gradient problem that was present in MobileNetV1 (*Sandler, M. et al. 2018*).

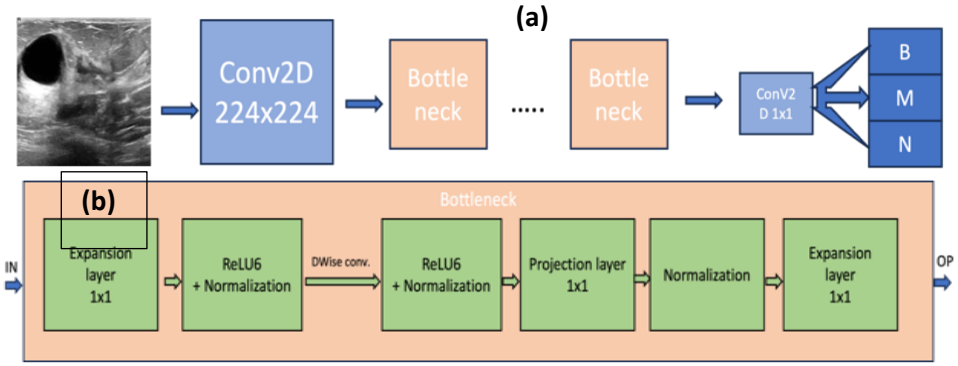


Figure 4: **a)** This is a diagrammatic representation of network architecture **b)** The internal structure of a bottleneck layer.

Figure 4a represents the architecture of MobileNetV2 network. Figure 4b gives a detailed view of a bottleneck layer. In this layer, we can see the reverse residual approach, where the residual layers connect the bottlenecks (the expansion layer at the beginning and at the end of the bottleneck). Here, the expansion layer uses a 1x1x1 kernel and expands the feature space for all channels and ReLU6 and normalization are applied, this expanded layer is then subjected to depthwise convolution and followed by pointwise convolution thereby reducing the feature space of the output layer, the last expansion layer once again expands the input received after pointwise convolution to increase the feature space back to the normal shape (*Sandler, M. et al. 2018*). MobileNetV2 use 30% fewer parameters and reduced operations by 2 when compared to MobileNetV1(previous version called MobileNet) while maintaining the same accuracy, making it more efficient than MobileNetV1 (*Sandler, M., et.al. 2018*).

➔ ***Transfer learning***:

The MobileNetV2 used in this research is a pre-trained network from the PyTorch framework. This pre-training is called Transfer Learning, in which the model is trained on large widely available data, like the ImageNet dataset, the model layers are frozen (so that the weights are fixed) and are fine-tuned. The feature extraction layers along with the last classification layer get trained on new data. This is a useful approach of CNN, to pre-train models so that the model learns fast on the new training data, with lower training time (*Zhou, S.K., et.al. 2023*). Since the data used in this research is small, a pre-trained MobileNetV2 model is used.

2.3. Image generation:

As proposed by *Frid-Adar, M. et al. (2018),* synthetic image generation for extending the dataset for training using DCGANs, has shown improved system performance. In this research, the data generation is for live lesions. The authors have compared image augmentation by classical techniques and diverse image generation by GANs. The model's sensitivity and specificity were used as metrics for performance evaluation. The model performed at 78.6% sensitivity and 88.4% specificity with the classical data augmentation and with DCGAN, the model performance increased to 85.7%. for sensitivity and 92.4% specificity, thus concluding that the system was improved by 7% using synthetic data augmentation.

Zhang, L. et al. (2021) demonstrated that DCGAN generates higher-quality images than vanilla GAN and requires less computational time compared to another GAN variant, ProGAN. In their research, the authors utilized High Content Screening (HCS) images of Human Monocytes treated with bacteria and subsequently induced with various FDA-approved (Food and Drug Administration) drugs to analyze the phenotypic profile. The classification system employed was the one-class SVM, which, when trained with only real images, achieved an accuracy of 97%. The accuracy increased to over 98% when the system was trained with synthetic images produced by DCGAN. The classification results reflected the types of effects the FDA-approved drugs had on the human monocytes induced with bacteria.

In a study conducted by *Alrashedy, H.H.N et al. (2022),* the researchers utilized a Brain MRI image dataset. They performed a comparative study between vanilla GAN and DCGAN and

classified the images using the ResNet152V2 model. The model yielded an overall accuracy of 94.84% for images generated by vanilla GAN and an overall accuracy of 99.09% for those generated by DCGAN, thereby demonstrating the superior efficiency of DCGAN. Hence DCGAN was opted for this research as it demonstrates good system performance for synthetic data augmentation. GANs and DCGANs structure is explained below.

Goodfellow I., et al. (2014) paved the foundation for Generative Adversarial Networks (GANs), which have greatly evolved since then. The working of GANs, as explained by *Goodfellow, I., et al, (2014)* is as follows:

GANs are neural networks that consist of two models, a Generator (G) and a Discriminator (D). G captures the data distribution and D is used to determine if the image generated by G is from the training set or a fake one. G gets trained based on the output of D, such that, the more D makes mistakes in identifying the generated images as fake, the better G gets. *"The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation." (Goodfellow, I., et al., 2014)*

GANs are made up of two neural network models. As Depicted in Figure 5, the G is fed with a noise vector and using this, it trains to generate images to match the training images. This training of G is achieved based on the output of D. G keeps generating images and D examines and determines the probability that the image generated by G is the training one (output as 1) or a fake one (output as 0). This process keeps repeating until a point when G can generate images so close to the original ones that D determines the fake image is real. As this is achieved, the loss incurred by D which is initially High is lowered and the loss by G increases and the loss converges.

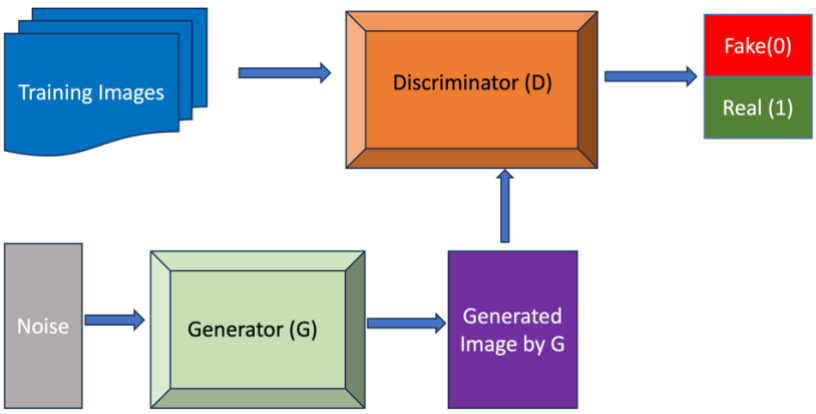


Figure 5: working of a GAN.

Figure 6: New images (non-existing faces) generated by GANs as published by *Karras, T., et al., (2017)*

Since the emergence of GANs, they have been used in multiple applications such as Object detection, image inpainting, video generation, and image generation (shown as an example in Figure 6) etc., (*Alqahtani, H., et.al. 2019*) One such application is data augmentation. There are many traditional image augmentation techniques such as flip, rotation, and scaling. However, the data obtained with these lacks' variety, hence less scope for improvement of the learning model. Therefore, GANs can be used to generate complex and diverse images and help solve this issue(*Frid-Adar, M. et al. 2018*).

The GANs work based on a two-player min-max game and the equation on which the GAN works is provided below, in equation (1):

$$\min_G \max_D F(D,G) = \mathbf{E}_{x\sim pr}[\log D(x)] + \mathbf{E}_{z\sim pz}[\log(1 - D(G(z)))] \quad (1)$$

Here, $\min_G \max_D F(D,G)$ is the min-max function where *pr* is the real data distribution on which the discriminator D is trained, *pz* is the fake data distribution on which the generator is trained. D is trained on real data D(x) and is expected to maximize the value for $E_{x\sim pr}[logD(x)]$, meaning to provide a high probability close to 1, and G(z). D is also expected to reduce the term $E_{z\sim pz}[log(1-D(G(z)))]$ as $D(G(z))$ is data distribution of fake data where G wants to maximize the $G(z)$ as close to the real data, hence D should minimize this term by giving low probability close to 0, using *log(1-D(G(z))) (Goodfellow I., et al., 2014)*.

➔ **DCGAN**:

Deep Convolutional GAN (DCGAN) is one of the important variations of GAN and it forms the basis of many other variations (*Öcal, A. and Özbakır, L., 2021*). The discriminator (D) is made up of convolutional layers and the generator (G) is made up of convolutional-transpose layers (*Radford, A., et.al, 2015*).

Transpose convolution is the reverse of a convolution, where the input is subjected to up-sampling. This is achieved through the regular rearrangement of various intermediate feature representations that are derived from performing several convolutional processes on the input feature maps. Each transpose convolution can be considered as a sim of multiple convolutional actions. It can be said that the input to the transpose convolutional layer is the output of the convolutional layer.(*Gao, H., et.al. 2019).*

The standard activation functions used in both the networks are LeakyReLU (for all initial and intermediate layers), tanh for the final layer of the generator, and sigmoid for the final layer of the discriminator (*Radford, A., et al. 2015*). Neural Networks are made up of multiple layers of neurons, and they should consist of at least two layers. The neurons from output of one layer are passed as an input to another layer. Based on the best-desired accuracy, the number of layers can be updated. A neural network without an activation function acts as a linear regression model and produces a linear output, with low learning. Therefore, for the model to learn better and to understand the complexity of the data, activation functions (non-linear) should be used. The activation functions used here are Sigmoid (transforms output values between 0 and 1), tanh (transforms output values between -1 and 1) and LeakyReLU (Transforms output values between slightly below 0 and 1). These activation functions also help in optimizing the Gradient (error loss) of the neural network which updates the weights of the neurons. (*Sharma, S., et.al. 2017*).

Another important step is Batch Normalization which normalises the input to mean 0 and constant standard deviation. Since the normalization happens on mini batches of inputs, it is called Batch Normalization. By normalizing the input, the activations are in check and the "activation exploding"(vanishing gradients) issue is avoided which reduces bias and results in better generalization. Hence this is used in every layer (except the last layer). As the last layers have separate activation layers and also batch normalization is not used for the output as it is applicable for input (*Bjorck, N., et.al. 2018*).

The algorithm of the working of the generator and discriminator is given in Figure 7. The first and the last layers of both networks are those that form the basic DCGAN. As per requirement, the layers can further be added, and both networks can be fine-tuned to obtain the desired output (*Radford, A., et.al, 2015*).

for epoch==1 to n
    Train generator (**G**) to produce data (z) of M-sized batch with probability distribution Pz (**G**(z))
    Train discriminator (**D**) on M M-sized batch of data from training data (x) with probability distribution Pd (**D**(x))
    Train the **D** on the M-sized batch of z data produced by the generator (**D**(**G**(z)))
    Calculate **G** loss using Binary Cross Entropy
    Calculate **D** loss using Binary Cross Entropy in total for **D**(x) and **D**(**G**(z))
    Calculate gradients using the loss for **G**
    Calculate gradients using the loss for **D**
    Optimize gradients using Adam optimizer for **G**
    Optimize gradients using Adam optimizer for **D**
    Increment epoch by 1
End when epochs>n

Figure 7: Base Algorithm for DCGAN used in this research.

3. Methodology

3.1. Data Acquisition and Data Pre-processing:

The Dataset was downloaded from the university page of Professor Aly Fahmi (https://scholar.cu.edu.eg/?q=afahmy/pages/dataset) and cited as per the request mentioned on the website. (*Al-Dhabyani W et al.,2020)*

The dataset contains 780 ultrasound images of breast and are of three categories Benign, Malignant and Normal(437 Benign, 210 Malignant and 133 normal; samples shown in Figure 8). Data pre-processing involves resizing the images to 224*224 and rescaling for normalization. (*Addala, V. (2023))*. The images are transformed by using normalize with mean = [0.485, 0.456, 0.406] and std=[0.229, 0.224, 0.225], which is required as per usage guidance by PyTorch. *PyTorch (2024)*

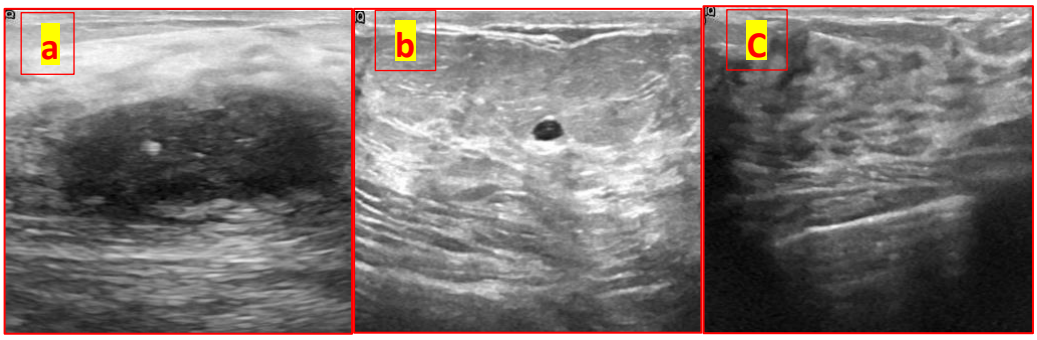


Figure 8: The figure shows the sample dataset, **a)** Ultrasound of a benign tumour in the breast **b)** Ultrasound of a malignant tumour in the breast **c)** Ultrasound of a normal breast with no tumours.

3.2. Software and library used:

The code was written in Google Colaboratory Pro+ (NVIDIA V100 GPU and high RAM , available only on Pro+, as the number of epochs was very high for image generation) and the deep learning frameworks used in this research are PyTorch (torch library), TensorFlow and Keras.

3.3. MobilenetV2 from Pytorch framework:

In this research, for the classification of breast cancer tumour type, MobilenetV2 of PyTorch framework has been selected. There are many frameworks, but the most commonly used frameworks are TensorFlow and PyTorch. The existing application as mentioned in the literature review, was built using MobilnetV2 in TensorFlow. In this study, we opted for Pytorch due to the user-friendliness of building models and due to better execution time as demonstrated by *Novac, O.-C. et al. (2022)*. PyTorch models can be converted to mobile version under PyTorch Mobile, which works without any additional internet connection and on the device remotely.

The PyTorch model was loaded using the code below - The model used is pretrained, which works on transfer learning (pre-trained on ImageNet dataset)(*Zhang, X., et.al. 2022*), and requires the input image to be of a minimum 224x224 size and it takes in RGB, 3 channels of the image. As mentioned in the Data Pre-processing section, images are required to be normalized and then fed into the model (*PyTorch, 2024)*. Using the above framework, the existing system was replicated with 20 epochs, giving 86% as the best accuracy. This system was then updated as System 1.

```
import torch
model = torch.hub.load('pytorch/vision:v0.10.0', 'mobilenet_v2', pretrained=True)
model.eval()
```

3.3. System 1

In the next step, the model was then updated to work with K-Fold Cross-validation. This updated model will be referred to as the System 1. System 1 was tuned and tested with different K values for the folds, and it performed the best with K=10. Hence, we can say that System 1 was efficient at 10-fold. Accuracy and confusion matrix were taken as performance metrics.

3.4. DCGAN for new images

DCGAN was programmed using TensorFlow and Keras, as TensorFlow gives better performance, and we require this only for data augmentation for training. Both the Generator and Discriminator are built separately. Both models follow sequential layering, and the model works in a step-by-step manner. Both networks have batch normalization layers to normalize the inputs for each layer. The activation used is LeakyReLU. LeakyReLU is a standard activation function for the Discriminator. After tuning for different activation functions, like ReLU, LeakyReLU seemed to be the best fit for the Generator. Hence, both models use LeakyReLU activation, with a common learning rate of 0.001.

The first layer in the Generator network is the Dense layer to which a noise vector of 100 is passed. The noise vector is 100 as per standards (*Radford, A., Metz, L. and Chintala, S., 2015*). As depicted in Figure 9, the inner layers used are 2d transposed-convolutional layers. These layers perform the transpose convolutional process, i.e., map the input tensor to a larger output tensor, which is how we can get a fake image of the original size 224x224 at the end of the last layer (*Zheng, S., 2021*). The size for the noise vector to be transpose convoluted at the initial layer is 28x28 with a depth of 256. This increases for every layer and is followed by 28x28 at depth 128, 56x56 at depth 64, 112x112 at depth 32 and finally flattened to match the original image size to 224x224 in the last layer and the output is of size 224x224.

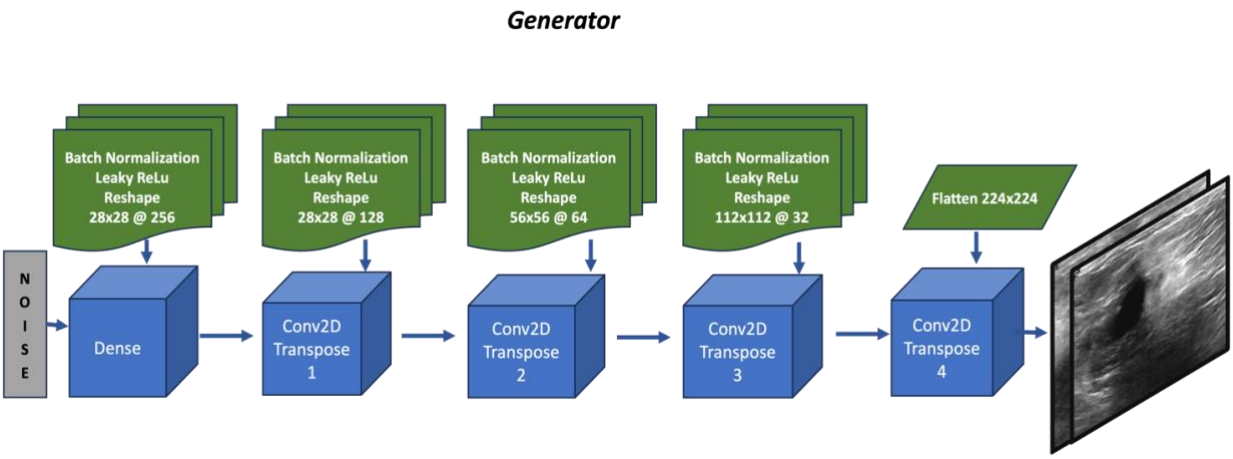


Figure 9: Diagrammatic representation of the Generator network used in this study.

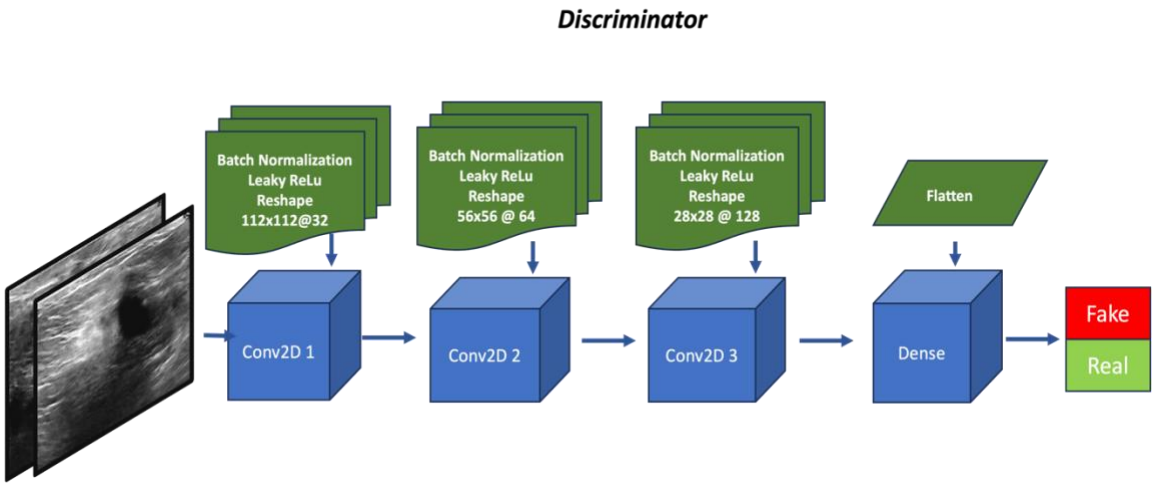


Figure 10: Diagrammatic representation of the Discriminator model used in this study.

The Discriminator uses 2d Convolutional layers. It is trained with the original data and then the images generated by the Generator are passed. The Discriminator model is layered as per the depiction in Figure 10. The layers convolute, i.e., extract the features and based on the feature extraction of the trained data, the Discriminator understands the feature extraction of the generated data. The last layer is Dense layer, and the data is flattened, with a sigmoid activation function.

To balance the data, all three classes should be of the same size. Since the Benign class has the highest number of images (437), the other two classes had to be balanced with the same number of images. 227 images were generated for the malignant class (which had 210 images originally) to make it to 437 and 304 images were generated for the normal class (which had 133 originally) to make it to 437. For the malignant class, the number of epochs required to produce the desired images was 4200 and for the normal, the number of epochs was 2500. The batch size used is 150 for the malignant class and 100 for the normal class. This was achieved after tuning.

Both the models were fine-tuned by adding and removing layers, by changing the number of epochs and the batch size. The tuning was stopped based on the visual appearance of the images along with the reduction in the loss calculation by the discriminator.

3.5. Using the new data for classification.

The System 1 with new data will be referred to as System 2, from this point onwards in this thesis. In System 2, the new data was only added to the training, and the testing was done solely on existing data in the 10-fold cross-validation.

3.6. Performance metrics used:

The confusion matrix provides the details of the true data vs the predicted data. It gives an in-depth analysis of how the system has predicted the data, how many instances of a particular class were identified as that class and how many were wrongly predicted as other classes. The confusion matrix has four terms namely, True Positives (TP) (correctly identified instances), True Negatives (TN) (correctly identified instances that belong to other classes), False Positives (FP) (incorrectly identified instances from other classes as a particular class for which we are analysing the matrix) and False Negatives (FN) (incorrectly identified instances from a particular class for which we are analysing the matrix as a different class). Figure 11 gives the depiction to analyse the confusion matrix for the three classes of data used in this research.

Accuracy is the most common metric used for system performance evaluation and it is measured as the number of predicted instances for a class divided by the total number of instances multiplied by 100 (*Baratloo, A., et.al., 2015*)

Accuracy =(TP+TN/(TP+TN+FP+FN))*100

Sensitivity and Specificity are the most commonly measured metrics in medical systems to evaluate performance (*Addala, V. 2023*). They are calculated from the confusion matrix and are defined as given below:

Sensitivity: It is the measure of a system's ability to correctly identify the presence of a specific class of tumour (*Swift, A., et.al, 2019*).

Sensitivity = TP / (TP +FN) ….(*Addala, V. 2023*).

Specificity: It is the measure of a system's ability to correctly identify the absence of a specific class of tumour. (*Swift, A., et.al, 2019*).

Specificity = TN / (TN+FP) ….(*Addala, V. 2023*).

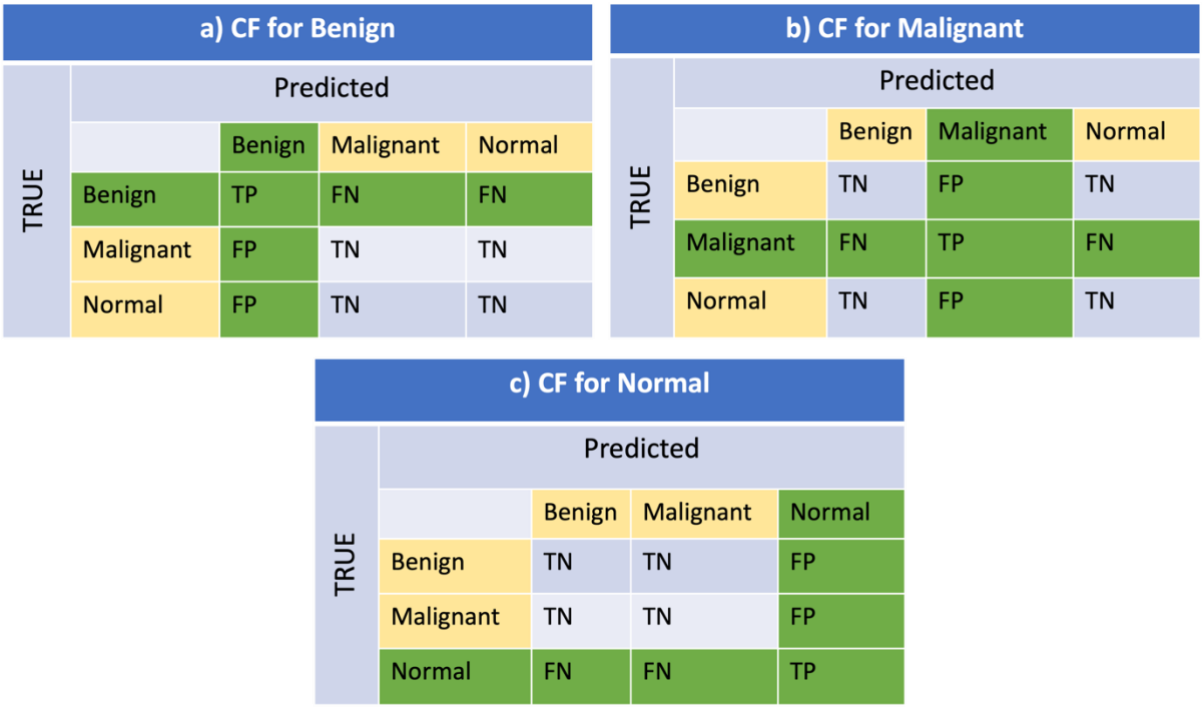
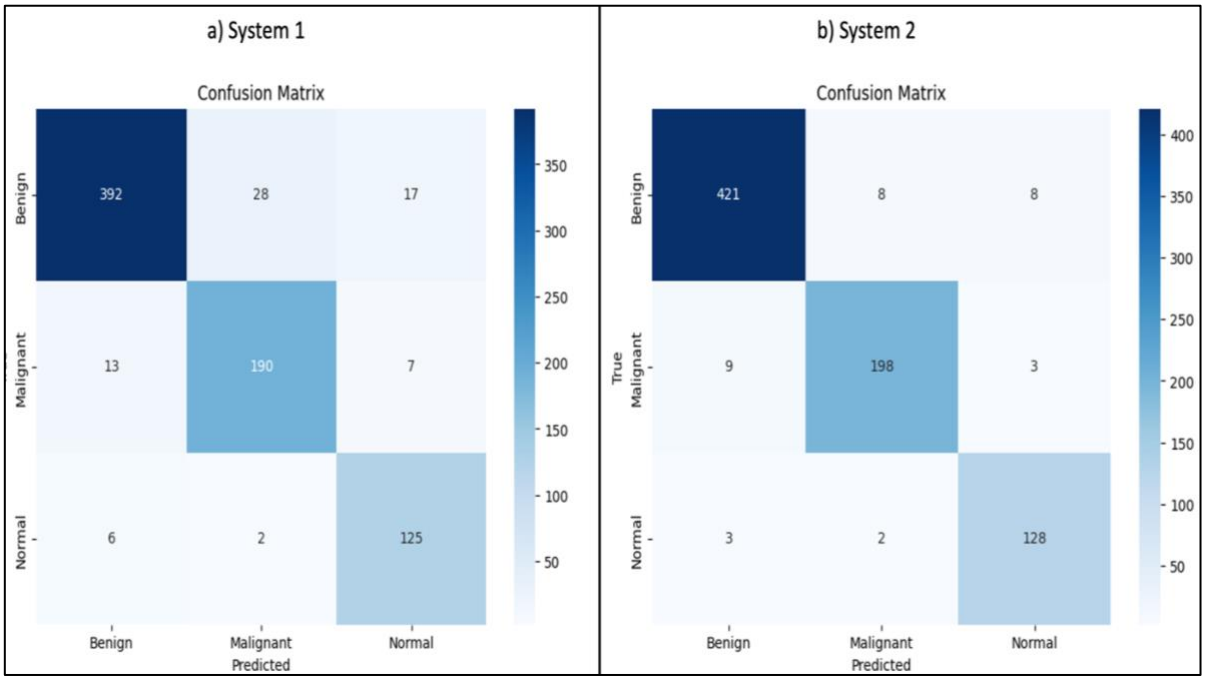


Figure 11: **a)** Confusion matrix calculation for benign class **b)** Confusion matrix calculation for malignant class **c)** Confusion matrix calculation for normal class

4. Results and Discussion

For the classifier MobileNetV2, the metrics used here are the Train and Test Accuracy of the whole system, Sensitivity and Specificity of every class derived from the Confusion matrices. The results are shown in Table 1. The best results for both System 1 and System 2 were obtained at 20 epochs beyond which both the systems had lower test accuracy, and higher train accuracy indicating higher variance (overfitting).

The confusion matrices for each system are presented in Figure 12. When comparing System 1 with the Existing System, there is a slight improvement for the benign class, with sensitivity increasing from 0.86 to 0.89 and specificity from 0.93 to 0.95, suggesting that System 1 is more efficient at distinguishing benign from non-benign tumours. For the malignant class, however, there is a decrease in sensitivity from 0.95 to 0.90, indicating a reduced ability of System 1 to identify malignant tumours compared to the Existing System. Despite this, System 1 has shown a significant improvement in specificity from 0.81 to 0.95, which implies better identification of non-malignant tumours. Nonetheless, this trade-off between decreased sensitivity and increased specificity may not be favourable, as accurate identification of malignant tumours is crucial. In the case of the normal class, System 1 exhibits a substantial increase in sensitivity from 0.56 to 0.93, with a marginal decline in specificity from 0.99 to 0.96, indicating superior performance over the Existing System in recognizing normal cases. Overall, System 1 shows enhanced performance for benign and normal classes but falls short for the malignant class compared to the Existing System.



c) Existing System

Predicted \ Actual	Normal	Benign	Malignant
Normal	14	1	0
Benign	2	77	2
Malignant	9	12	39

Figure 12: **a)** Confusion Matrix of System 1, **b)** Confusion Matrix of System 2 and **c)** Confusion Matrix Existing System

	Existing System			***New System with K fold(System 1)***			***New System with K fold+ New images (System 2)***		
	B	**M**	**N**	**B**	**M**	**N**	**B**	**M**	**N**
Sensitivity	0.86	0.95	0.56	0.89	0.90	0.93	0.96	0.94	0.96
Specificity	0.93	0.81	0.99	0.95	0.95	0.96	0.97	0.98	0.98
Accuracy Train	95.65			98.713			98.958		
Accuracy Test	87.50			93.432			94.929		

B = Benign ; M = Malignant ; N = Normal

Table 1: Table comparing the performance metrics of the Existing system, System 1 and System 2

Considering the metric values of System 2, there is an observed overall increase in both sensitivity and specificity compared to System 1. Sensitivity for the benign class has risen from 0.89 to 0.96 in System 2, and specificity has improved from 0.95 to 0.97. For the normal class, sensitivity has increased from 0.93 to 0.96, and specificity has marginally improved from 0.96 to 0.97 in System 2. These enhancements demonstrate that System 2 outperforms both System 1 and the existing system in differentiating benign from non-benign, and normal from non-normal cases. In the malignant class, System 2 shows increased sensitivity, from 0.90 to 0.94, and specificity, from 0.95 to 0.97, indicating superior performance in identifying malignant from non-malignant tumours compared to System 1 and the existing system.

The Train accuracy has increased drastically in System 1 from 95.65% to 98.713% when compared to the Existing system. The Train accuracy has marginally increased from 98.713% to 98.958% in System 2 when compared to System 1. System 1 shows a drastic increase in Test accuracy compared to the Existing system, from 87% to 93.432%, indicating reduced variance (overfitting) in System 1 when compared to the Existing system. The Test accuracy in System 2 has increased from 93.432% to 94.929%, a ~1.5% increase in Test accuracy when compared to System 1. This indicates that System 2 has lesser variance than both System 1 and the Existing system.

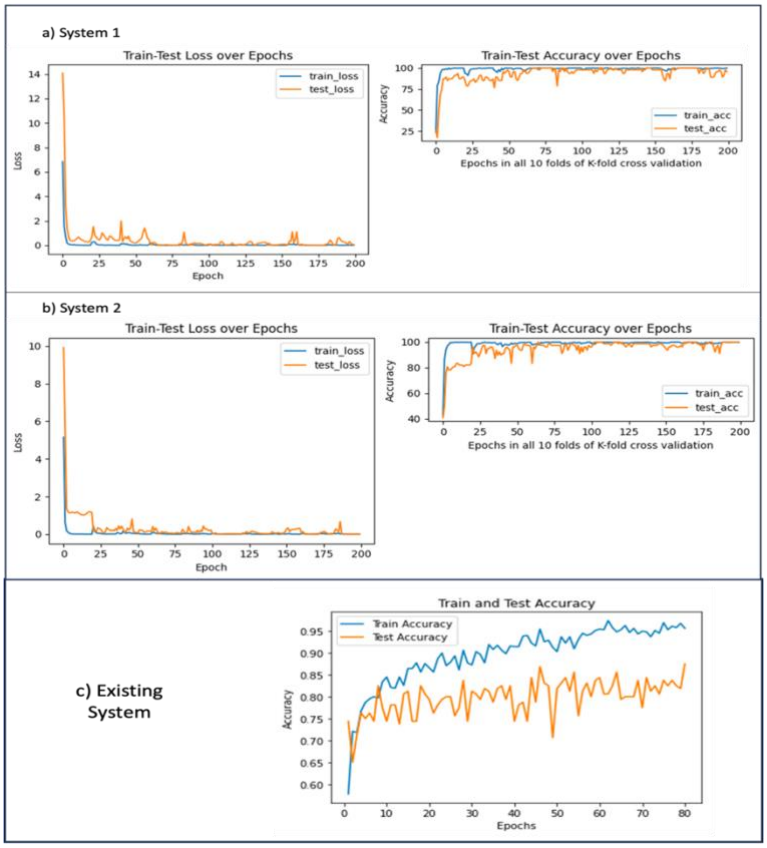


Figure 13: a) Tran-test loss over total epochs for System 1, **b)** train-test accuracy over total epochs for System 2, **c)**train-test accuracy of existing system

The Train-Test Loss and Accuracy graphs for Systems 1 and 2, along with the Train-Test Accuracy graphs for the Existing System, are depicted in Figure 13a,b,c. Systems 1 and 2 implement 10-fold cross-validation, where each fold comprises 20 epochs; thus, for 10 folds, the total is 200 epochs. Both systems exhibit a pronounced decrease in loss from epochs 0-25, indicating rapid learning in the initial training phase. While there are minor fluctuations in test loss compared to train loss for both systems, these are negligible, suggesting effective generalization of the data. The accuracy graphs demonstrate a quick increase in train and test accuracies across all systems. However, the Existing System shows a widening gap between train and test accuracies as training progresses, hinting at overfitting due to the high train accuracy but lower test accuracy. Conversely, Systems 1 and 2 display a trend toward convergence of train-test accuracies at various training points, which suggests that they are more robust against overfitting compared to the Existing System.

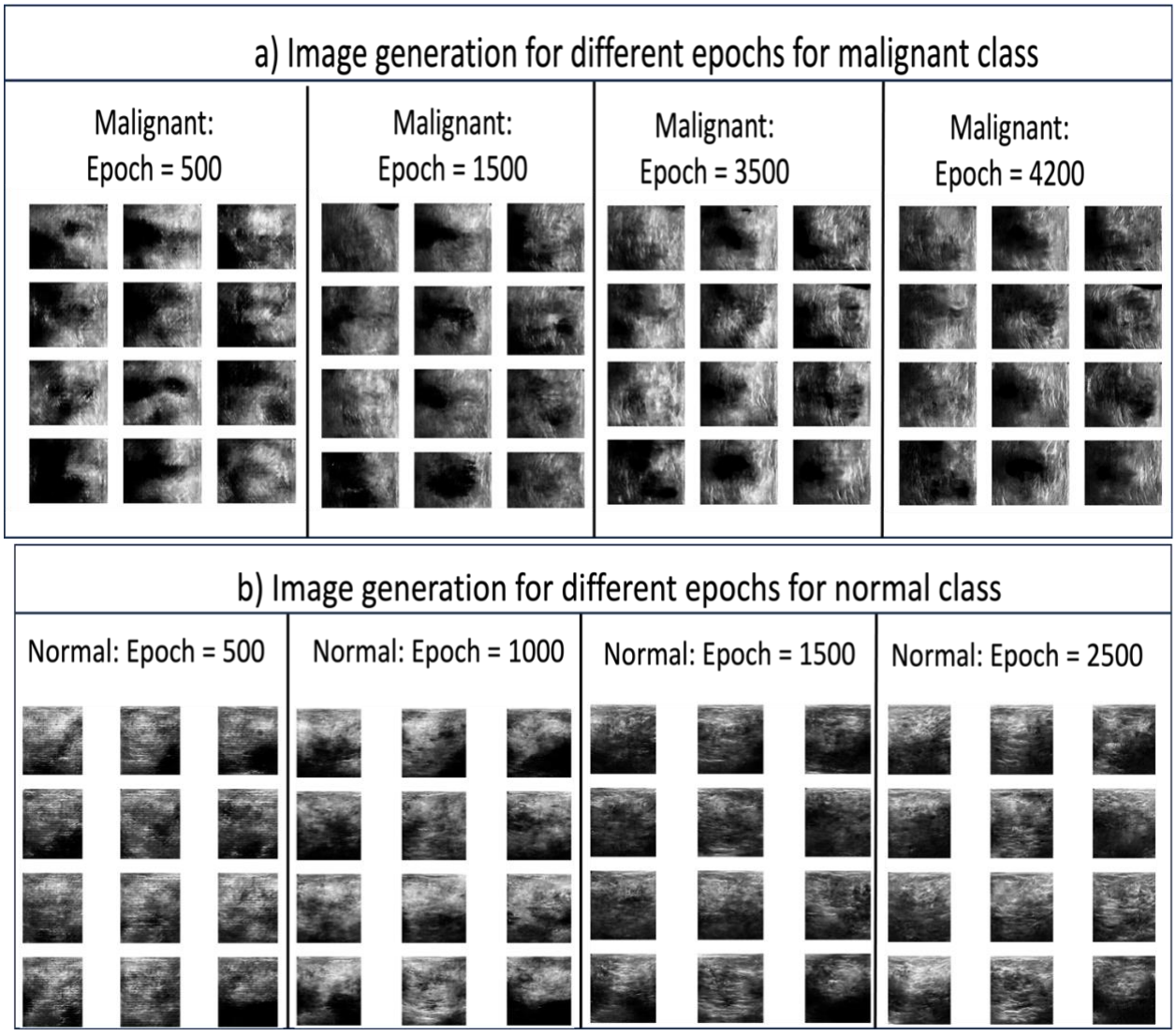


Figure 14: a) comparison of generated malignant class images for epochs 500, 1500, 3500, 4200 that improved as the number of epochs increased **b)** comparison of generated normal class images for epochs 500, 1000, 1500, 2500 that improved as the number of epochs increased.

The image generation results are shown in Figure 14. The image generation was done until the images were visually interpretable and until the generator and discriminator loss were close to each other with values close to 0. Figure 14a and Figure 14b show the improvement in the images for malignant and normal class image generation. For the malignant class, as shown in Figure 14a, the best results were obtained at 4200 epochs. We can see that for epochs 500 and 1500, the images have a large amount of noise and for 2500, 3500 and 4200 the images have a noise reduction, resembling the original dataset. For the normal class, the best images obtained were at epoch 2500. In the Figure 14b, the images in epochs 500 and 1000 have high distortion. As the epoch increases there is more stability and better image generation (epoch 1500, 2000 and 2500). The number of epochs retrains the generator to produce better images, thus decreasing Discriminator loss and making it decide that the generated images are real.

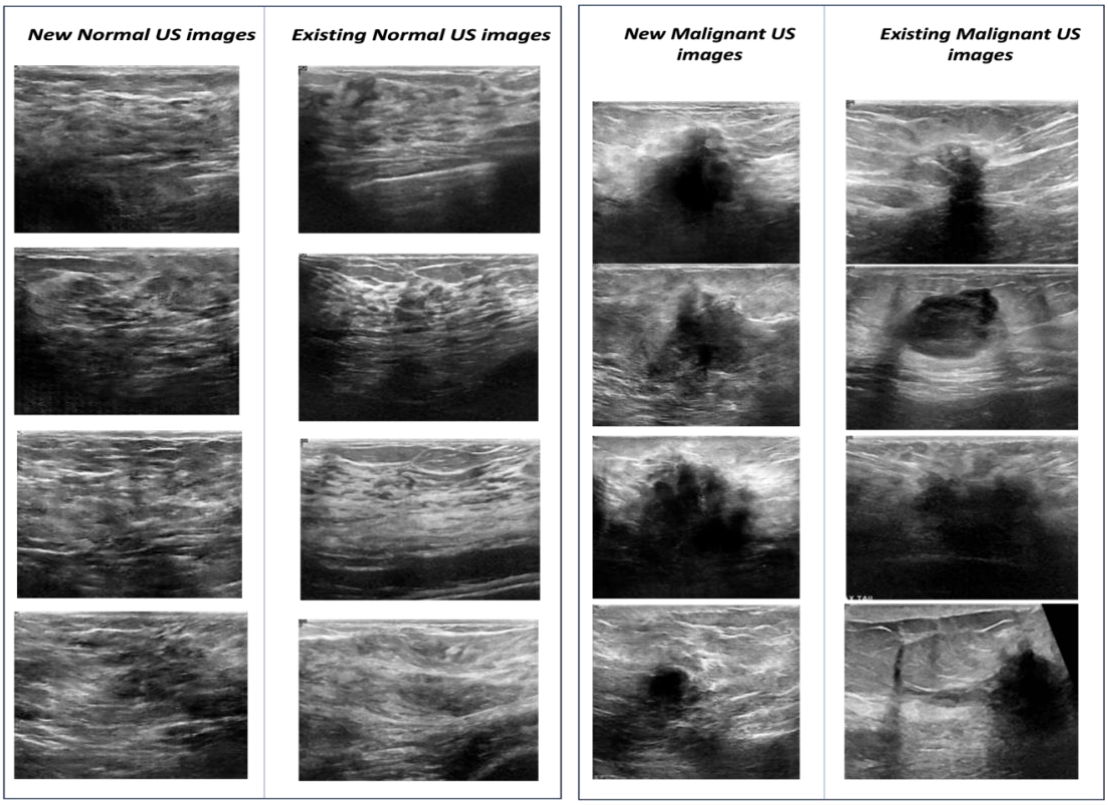


Figure 15: side-by-side comparison of new images and existing images in malignant and normal classes. respectively

The final images obtained after generating images are shown in Figure 15 and are compared with existing images. It can be observed that there is resemblance as well as diversity in the newly generated images, which are very similar to the original images.

The graph showing the losses for the generator and discriminator for the normal is presented in Figure 16a. The primary goal within the DCGAN framework is to minimize the discriminator's loss and keep it close to 0, while also ensuring the generator's loss is as close as possible to the discriminator's loss. According to the figure, there is an initial rapid decrease in loss for both neural networks, indicative of swift learning. The noticeable fluctuations suggest that the model

is actively learning. The discriminator's loss remains low, and the generator's loss has risen as the number of epochs increases (between 1500 to 2500). The increase in loss is very little, this suggests that the discriminator may be slightly overfitting, and the generator is not learning marginally. Overall, the model performs well with the generator generating data close to training data with slight overfitting.

The graph depicting the losses for the generator and discriminator corresponding to the malignant class is shown in Figure 16b. The figure indicates an initial sharp decline in loss for both neural networks, signifying rapid learning. The significant fluctuations imply active learning within the model. The discriminator's loss remains low, whereas the generator's loss has marginally increased as the number of epochs grows, particularly between 2500 to 4200 epochs. This rise in loss is minimal. However, the sudden spikes observed between 2800 to 3900 epochs point to overfitting. Overall, the model is performing well, with the generator creating data closely resembling the training data, albeit with a higher degree of overfitting in comparison to the normal class.

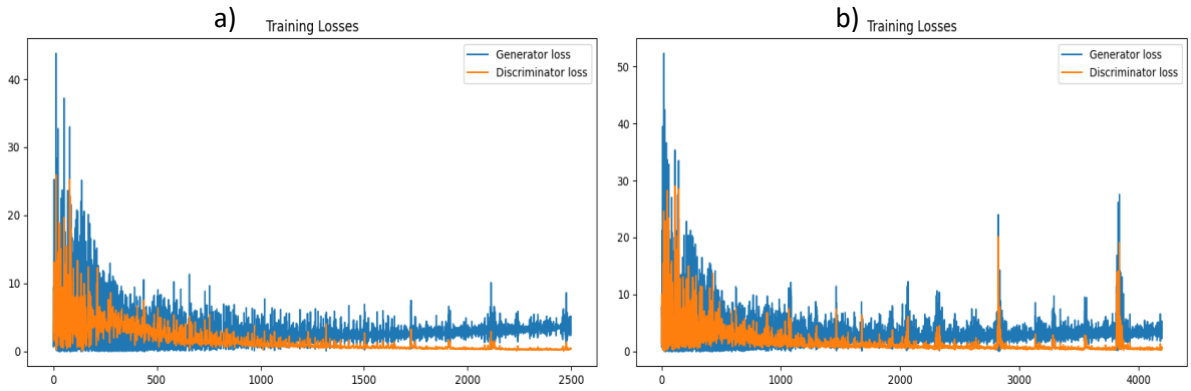


Figure 16: a) The graph of Generator-Discriminator loss vs number of epochs for generated normal class images,, b) The graph of Generator-Discriminator loss vs number of epochs for generated malignant class images.

5. Conclusion, limitations and future work:

The primary hypothesis of this project was to determine whether synthetically generated images could enhance the performance of systems running on devices such as smartphones. It was observed that the model's accuracy improved from 87% to 94.929%. Additionally, the sensitivity and specificity for all classes increased, with a notable enhancement for the 'normal' class, which previously suffered from a lack of data. This confirms that augmenting the dataset with images generated by DCGAN can indeed improve the system's performance, thereby validating the hypothesis.

One limitation of this project was the absence of medical expertise to verify the medical accuracy of the generated images (intervention of medical expertise is required only for evaluation purposes and is not required in any other step in this project). Therefore, for future work, it is recommended that the generated images undergo medical review. Additionally,

images for all classes could be generated and the system's performance evaluated comprehensively, a goal that was not attainable in this project due to time constraints. A future task involves evaluating the quality of the generated images with deep learning techniques to refine the DCGAN configuration and reduce the slight overfitting. Additionally, applying and comparing more sophisticated GAN variants could further enhance the quality of the images produced. By addressing this, the gap in medical research due to the limitation of lack of data can be closed.

6. References (~10)

1. Addala, V. (2023) 'Breast Ai: Low cost, explainable artificial intelligence based app for efficient diagnosis of breast cancer in developing areas', *2023 IEEE 3rd International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB)* [Preprint]. doi:10.1109/iceib57887.2023.10170357.
2. Albawi, S., Mohammed, T.A. and Al-Zawi, S., 2017, August. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)* (pp. 1-6). Ieee.
3. Al-Dhabyani W, Gomaa M, Khaled H, Fahmy A.(2020) Dataset of breast ultrasound images. Data in Brief. 2020 Feb;28:104863. DOI: 10.1016/j.dib.2019.104863.
4. Alqahtani, H., Kavakli-Thorne, M. and Kumar, G. (2019) 'Applications of generative Adversarial Networks (Gans): An updated review', *Archives of Computational Methods in Engineering*, 28(2), pp. 525–552. doi:10.1007/s11831-019-09388-y.
5. Alrashedy, H.H.N., Almansour, A.F., Ibrahim, D.M. and Hammoudeh, M.A.A., 2022. BrainGAN: brain MRI image generation and classification framework using GAN architectures and CNN models. *Sensors*, *22*(11), p.4297.
6. Arnold, M. *et al.* (2022) 'Current and future burden of breast cancer: Global Statistics for 2020 and 2040', *The Breast*, 66, pp. 15–23. doi:10.1016/j.breast.2022.08.010.
7. Baratloo, A., Hosseini, M., Negida, A. and El Ashal, G., 2015. Part 1: simple definition and calculation of accuracy, sensitivity and specificity.
8. Bevers, T.B. *et al.* (2009) 'Breast cancer screening and diagnosis', *Journal of the National Comprehensive Cancer Network*, 7(10), pp. 1060–1096. doi:10.6004/jnccn.2009.0070.
9. Bendersky2, E. (2003) *Depthwise separable convolutions for machine learning*, *Eli Benderskys website ATOM*. Available at: https://eli.thegreenplace.net/2018/depthwise-separable-convolutions-for-machine-learning/ (Accessed: 17 January 2024).
10. Bjorck, N., Gomes, C.P., Selman, B. and Weinberger, K.Q., 2018. Understanding batch normalization. *Advances in neural information processing systems*, *31*.
11. Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251-1258).
12. Dan, Q. et al. (2023) 'Ultrasound for breast cancer screening in resource-limited settings: Current practice and Future Directions', *Cancers*, 15(7), p. 2112. doi:10.3390/cancers15072112.
13. Davenport, T. and Kalakota, R. (2019) 'The potential for artificial intelligence in Healthcare', *Future Healthcare Journal*, 6(2), pp. 94–98. doi:10.7861/futurehosp.6-2-94.
14. Devolli-Disha, E. et al. (2009) 'Comparative accuracy of mammography and ultrasound in women with breast symptoms according to age and breast density', *Bosnian Journal of Basic Medical Sciences*, 9(2), pp. 131–136. doi:10.17305/bjbms.2009.2832.
15. Eroğlu, Y., Yildirim, M. and Çinar, A. (2021) 'Convolutional neural networks based classification of breast ultrasonography images by hybrid method with respect to benign, malignant, and normal

using mrmr', Computers in Biology and Medicine, 133, p. 104407. doi:10.1016/j.compbiomed.2021.104407.

16. Frid-Adar, M. *et al.* (2018) 'Synthetic data augmentation using GAN for improved liver lesion classification', *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)* [Preprint]. doi:10.1109/isbi.2018.8363576.
17. Gao, H., Yuan, H., Wang, Z. and Ji, S., 2019. Pixel transposed convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, *42*(5), pp.1218-1227.
18. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. *Advances in neural information processing systems*, *27*.
19. He, Y., Qian, J. and Wang, J., 2019. Depth-wise decomposition for accelerating separable convolutions in efficient convolutional neural networks. *arXiv preprint arXiv:1910.09455*.
20. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
21. Karras, T., Aila, T., Laine, S. and Lehtinen, J., 2017. Progressive growing of GANs for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.
22. Mehrotra, R. and Yadav, K. (2022) 'Breast cancer in India: Present scenario and the challenges ahead', *World Journal of Clinical Oncology*, 13(3), pp. 209–218. doi:10.5306/wjco.v13.i3.209.
23. Nie, D. et al. (2018) 'Medical image synthesis with deep convolutional adversarial networks', *IEEE Transactions on Biomedical Engineering*, 65(12), pp. 2720–2730. doi:10.1109/tbme.2018.2814538.
24. Novac, O.-C. *et al.* (2022) 'Analysis of the application efficiency of tensorflow and pytorch in Convolutional Neural Network', *Sensors*, 22(22), p. 8872. doi:10.3390/s22228872.
25. Öcal, A. and Özbakır, L., 2021. Supervised deep convolutional generative adversarial networks. *Neurocomputing*, *449*, pp.389-398.
26. PyTorch (2024) *MobileNetV2*, *PyTorch*. Available at: https://pytorch.org/hub/pytorch_vision_mobilenet_v2/#:~:text=Model%20Description&text=MobileNet%20v2%20uses%20lightweight%20depthwise,order%20to%20maintain%20representational%20power. (Accessed: 11 January 2024).
27. Radford, A., Metz, L. and Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.
28. Ragab, M., Albukhari, A., Alyami, J. and Mansour, R.F., 2022. Ensemble deep-learning-enabled clinical decision support system for breast cancer diagnosis and classification on ultrasound images. *Biology*, *11*(3), p.439.
29. Sandler, M. *et al.* (2018) 'MobileNetV2: Inverted residuals and linear bottlenecks', *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* [Preprint]. doi:10.1109/cvpr.2018.00474.
30. *MobileNetV2: The next generation of on-device computer vision networks,–Google Research Blog*. Available at: https://blog.research.google/2018/04/mobilenetv2-next-generation-of-on.html (Accessed: 21 January 2024).
31. Sedigh, P., Sadeghian, R. and Masouleh, M.T. (2019) 'Generating synthetic medical images by using gan to improve CNN performance in Skin cancer classification', *2019 7th International Conference on Robotics and Mechatronics (ICRoM)* [Preprint]. doi:10.1109/icrom48714.2019.9071823.
32. Sharma, S., Sharma, S. and Athaiya, A., 2017. Activation functions in neural networks. *Towards Data Sci*, *6*(12), pp.310-316.
33. Swift, A., Heale, R. and Twycross, A., 2019. What are sensitivity and specificity?. *Evidence-Based Nursing*.

34. Thomas, C., Byra, M., Marti, R., Yap, M.H. and Zwiggelaar, R., 2023. BUS-Set: A benchmark for quantitative evaluation of breast ultrasound segmentation networks with public datasets. *Medical Physics*.
35. WCRF International (2022*) Breast cancer statistics: World cancer research fund international, WCRF International*. Available at: https://www.wcrf.org/cancer-trends/breast-cancer-statistics/#:~:text=Breast%20cancer%20is%20the%20most%20commonly%20occurring%20cancer%20in,the%20most%20common%20cancer%20overall. (Accessed: 05 October 2023).
36. Zhang, L. and Zhao, L., 2021. High-quality face image generation using particle swarm optimization-based generative adversarial networks. *Future Generation Computer Systems*, *122*, pp.98-104.
37. Zhang, X., Zhou, J., Sun, W. and Jha, S.K., 2022. A Lightweight CNN Based on Transfer Learning for COVID-19 Diagnosis. *Computers, Materials & Continua*, *72*(1).
38. Zheng, S., 2021. Analysis of Generating Handwriting Based on GAN Model With Different Structures. In *2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)* (pp. 654-658). IEEE.
39. Zhou, S.K., Greenspan, H. and Shen, D. eds., 2023. *Deep learning for medical image analysis*. Academic Press.