Hanoi University of Science and Technology

School of Information and Communications Technology

# PROJECT REPORT

## Breast cancer detection using U-net model for image segmentation

**Course:** IT 3160E Introduction to Artificial Intelligence

| Authors: | Student ID: |
|---|---|
| Nguyen Tat Hung | 20235500 |
| Nguyen Vu Thuy | 20235562 |
| Nguyen Thanh Vinh | 20235576 |
| Le Nhat Hoang | 20235498 |

Hanoi, December 2024

# ABSTRACT

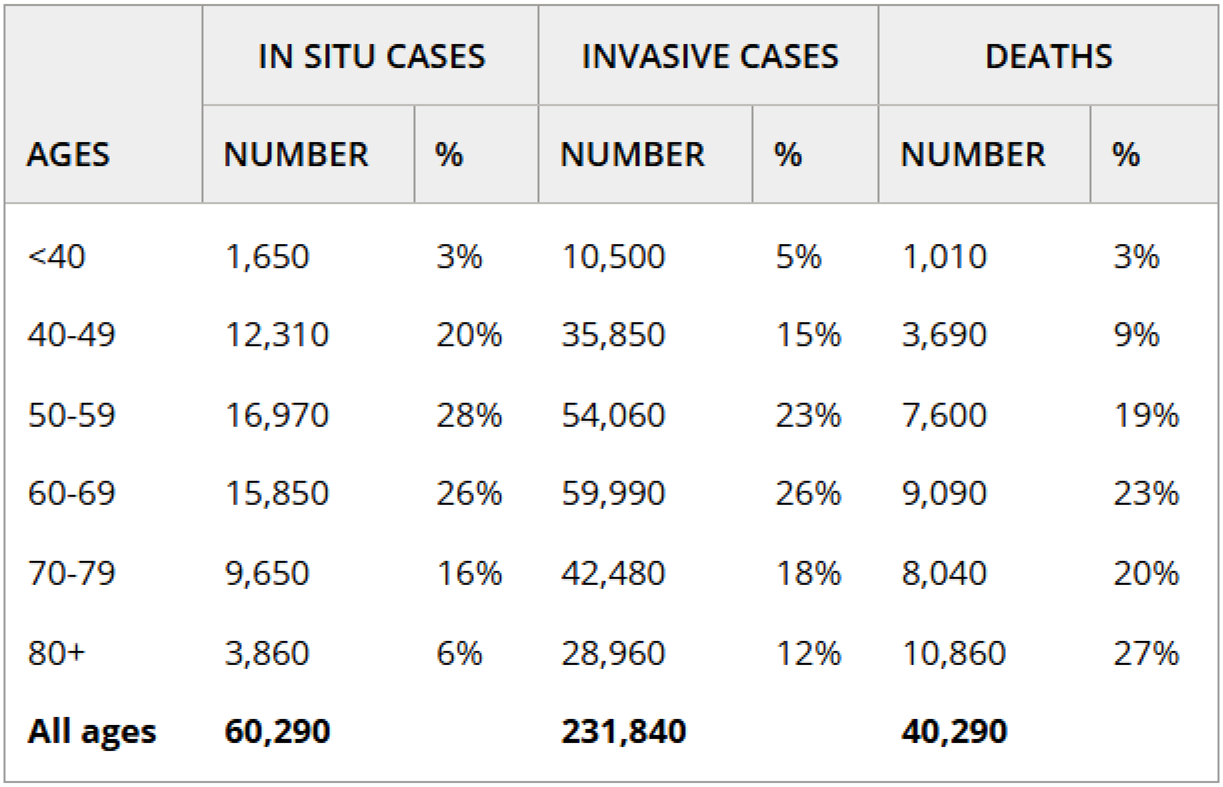
Breast cancer remains a leading cause of cancer-related deaths worldwide, highlighting the urgent need for advanced diagnostic methods. This study leverages the U-net deep learning model for segmenting ultrasound images to detect breast cancer, using a dataset of 780 images classified into normal, benign, and malignant categories. Preprocessing steps, including resizing and normalization, were applied to ensure optimal input for the model. The experimental results demonstrate U-net's superior performance, achieving high accuracy and low loss compared to SAM and ResNet models. This research underscores U-net's potential in enhancing early breast cancer detection and diagnostic accuracy, offering a promising AI-driven solution for personalized and efficient healthcare applications.

# Contents

# 1. Introduction

Breast cancer is a malignant tumor that develops from the cells of the breast, often beginning in the milk ducts or lobules. It is one of the most common cancers worldwide, affecting both women and, in rare cases, men. The disease can spread to other body parts through the lymphatic system or blood vessels, making early detection and treatment crucial.

Globally, breast cancer is the second leading cause of cancer-related deaths among women [01], with approximately 570,000 deaths reported in 2015 alone [02]. Every year, over 1.5 million women are diagnosed with the condition, highlighting the widespread nature of this disease [02] [03]. Early diagnosis and effective treatment have significantly improved survival rates, but challenges remain in ensuring that all cases are detected early enough to make a difference in outcomes.



|  | IN SITU CASES |  | INVASIVE CASES |  | DEATHS |  |
|---|---|---|---|---|---|---|
| AGES | NUMBER | % | NUMBER | % | NUMBER | % |
| <40 | 1,650 | 3% | 10,500 | 5% | 1,010 | 3% |
| 40-49 | 12,310 | 20% | 35,850 | 15% | 3,690 | 9% |
| 50-59 | 16,970 | 28% | 54,060 | 23% | 7,600 | 19% |
| 60-69 | 15,850 | 26% | 59,990 | 26% | 9,090 | 23% |
| 70-79 | 9,650 | 16% | 42,480 | 18% | 8,040 | 20% |
| 80+ | 3,860 | 6% | 28,960 | 12% | 10,860 | 27% |
| **All ages** | **60,290** |  | **231,840** |  | **40,290** |  |

Table 1: Estimated New Female Breast Cancer Cases and Deaths by Age, United States, 2015 [05]

Despite advances in imaging technology, detecting breast cancer early remains a complex challenge. Traditional methods such as ultrasound, mammography, and MRI have their limitations, particularly when it comes to sensitivity and resolution. Ultrasound, for example, is commonly used for screening and diagnosis, but its sensitivity can vary, often leading to false negatives or missed cancers. Additionally, ultrasound images are sometimes of lower resolution, which can make it difficult to identify smaller or more subtle tumors [04]. These limitations emphasize the need for more advanced and accurate tools to enhance early detection and improve outcomes for women worldwide.

# 2. Methodology

## 2.1. Preprocessing

### 2.1.1. Resizing

To ensure uniform input dimensions for the model, all images in the dataset are resized to a standard size of 256x256 pixels. Resizing images is essential because most deep learning models require fixed input dimensions. The 256x256 resolution is a commonly used size in image processing tasks as it is large enough to retain important features, yet not so large that it increases computational complexity or memory requirements. By resizing all images to the same size, we ensure that the model receives a consistent input, which improves the model's performance and training efficiency.

### 2.1.2. Nornalization

After resizing, the pixel values of the images are normalized by dividing each pixel value by 255, converting the pixel values from the range [0, 255] to [0, 1]. Normalizing the images is a critical step to improve the performance and stability of deep learning models. This step ensures that the model receives input data in a consistent range, which helps the optimization process. When pixel values are scaled to the [0, 1] range, the model is able to learn more efficiently and faster, as the gradients during training are less likely to become too large or too small (mitigating issues like gradient explosion or vanishing gradients). Additionally, this step allows the model to converge more quickly during training.

## 2.2. Segmentation

The model uses U-net for segmentation. U-Net is a deep learning architecture designed for image segmentation, particularly in medical imaging. It features a U-shaped structure with an encoder-decoder design, where the encoder captures context and the decoder restores spatial resolution. Skip connections between corresponding encoder and decoder layers help preserve fine details, improving segmentation accuracy. U-Net is known for its effectiveness with small datasets and has become widely used for tasks like tumor detection and organ segmentation due to its ability to generate precise, pixel-level predictions.

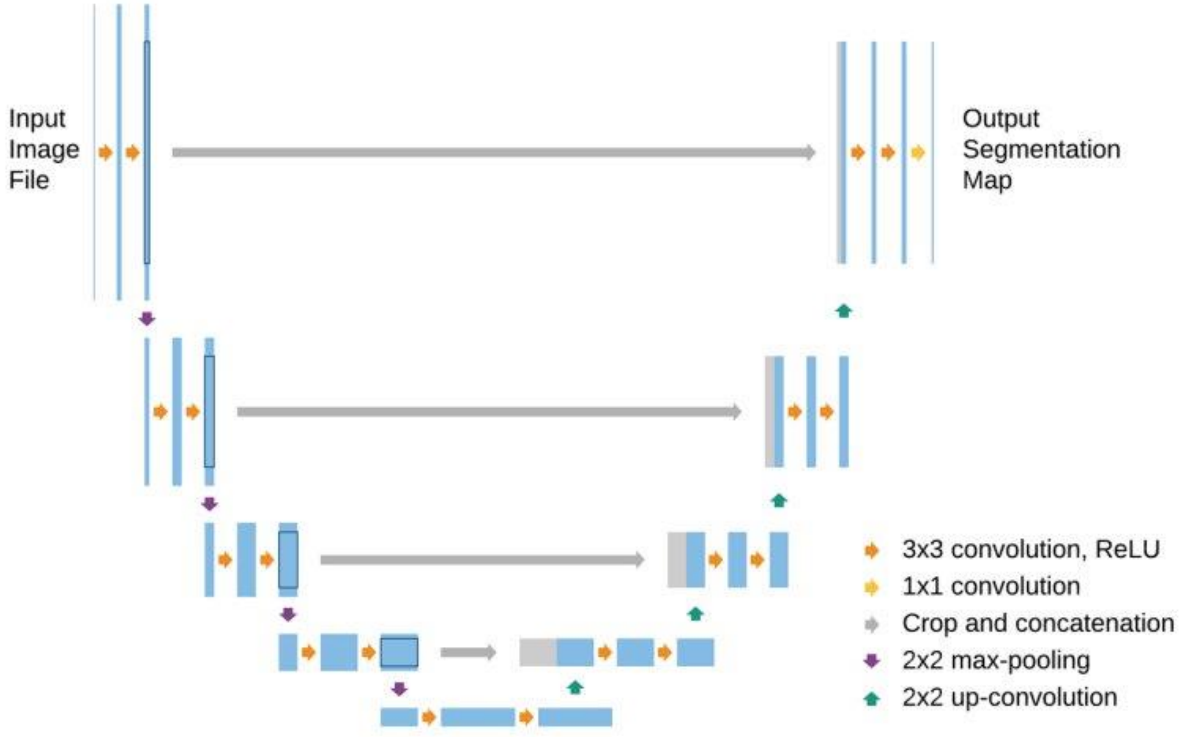
The encoder, also known as the contracting path, is responsible for extracting features from the input image. It does this by applying convolutional layers followed by pooling layers, which progressively reduce the spatial dimensions of the image. This process captures high-level information and provides context essential for accurate segmentation.

The decoder, or the expanding path, works to reconstruct the spatial dimensions of the feature maps. This part of the network generates a segmentation map by mapping the learned features back to the resolution of the input image. Through this process, the decoder ensures that the final output aligns with the dimensions of the original input while maintaining the necessary detail for segmentation.

One of the defining features of U-Net is the use of skip connections. These are direct links between layers in the encoder path and their corresponding layers in the decoder path. Skip connections help preserve spatial information that might otherwise be lost during the downsampling process in the encoder. By combining high-resolution features from the encoder with upsampled features in the decoder, the network achieves better accuracy in generating fine-grained segmentation results.

The ultimate goal of U-Net is to produce a segmentation mask, where each pixel in the input image is classified into one of the predefined classes. This makes U-Net particularly effective for tasks requiring pixel-level precision. For example, in medical imaging, U-Net can be used to detect tumors or other anomalies by accurately classifying each pixel within the image. Its ability to produce precise segmentation maps makes it an indispensable tool for applications in medical diagnostics and other fields requiring detailed image analysis.

### 2.2.1. U-net’s Network Architecture



(Fig. 1. U-net architecture. Each blue box corresponds to a multi-channel feature map [07])

The network architecture is illustrated in Figure 1. It consists of a contracting path (left side) and an expansive path (right side). The contracting path follows a typical convolutional neural network architecture. It consists of the repeated application of two 3x3 convolutions (unpadded convolutions), each followed by a ReLU activation function, and a 2x2 max-pooling operation with stride 2 for downsampling. After each downsampling step, the number of feature channels is doubled. The expansive path consists of upsampling the feature map, followed by a 2x2 convolution (termed "up-convolution") that reduces the number of feature channels by half, followed by concatenation with the corresponding cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU activation. Cropping is necessary due to the loss of border pixels during each convolution. At the final layer, a 1x1 convolution is used to map each 64-component feature vector to the desired number of output classes. In total, the network has 23 convolutional layers.

### 2.2.2. Advantage of U-net

First, the entire architecture does not use any fully connected layers. Working with deep learning, in typical end-to-end models, the penultimate layer is often a fully connected layer to integrate the extracted features for making predictions. However, in the U-Net architecture, the task of integrating features is handled by the second half of the "U", which eliminates the need for fully connected layers. This enables the network to accept inputs of any size.

Second, U-Net uses the padding method, which allows the architecture to fully segment images. This method is particularly important when segmenting images because, without it, the resolution could be limited by the GPU's memory capacity.

Moreover, U-Net can achieve good results with relatively small amounts of training data, that leads to high data efficiency. Also, with significant amount of flexibility, it works well for binary and multi-class segmentation tasks, especially with medical imaging and disease detection.

### 2.2.3. Application of U-net model into breast cancer recognition

The symmetric encoder-decoder architecture of U-Net model will be applied strictly into our project, which efficiently combines low-level spatial features with high-level semantic information through skip connections. This allows it to accurately segment complex structures, such as tumors, in medical images. In the context of breast cancer recognition, the model is trained to identify regions of interest-areas potentially indicative of malignancy-within breast tissue images.

The process begins with preprocessing breast cancer images to standardize inputs, followed by training the U-Net on annotated datasets. During inference, the model processes unseen images and outputs segmentation masks highlighting potential cancerous regions. The architecture and process are tailored specifically to the challenges of breast cancer imaging.

### 2.2.4. Algorithms with mathematical base

***The 2D Convolutional Layers (Applied in Encoder phase)***

The 2D convolutional layer in a neural network applies a series of filters (kernels) to the input image or feature map to produce an output feature map, capturing spatial patterns such as edges, textures, or other visual features, makes it foundational in modern computer vision tasks, including breast cancer recognition.

A 2D convolution operation can be expressed as:

$$Output(i,j) = \sum_{m=0}^{k-1}\sum_{n=0}^{k-1} Input(i+m, j+n) \cdot Kernel(m,n) \quad [08]$$

Where:

- $Input(i,j)$ is the pixel value of the input feature map at position $(i,j)$.
- $Kernel(m,n)$ is the value of the filter at position $(m,n)$.
- $k \times k$ is the size of the kernel (e.g., $3 \times 3$, $5 \times 5$).
- $(i,j)$ refers to the top-left corner of the receptive field, where the kernel is applied.

The convolution operation involves selecting a $k \times k$ receptive field (a patch of the input feature map) at a specific position.

For example, for a $3 \times 3$ kernel, the receptive field at position $(i,j)$ would be:

$$Receptive\ Field = \begin{bmatrix} Input(i,j) & Input(i,j+1) & Input(i,j+2) \\ Input(i+1,j) & Input(i+1,j+1) & Input(i+1,j+2) \\ Input(i+2,j) & Input(i+2,j+1) & Input(i+2,j+2) \end{bmatrix} \quad [09]$$

Multiply each element of the receptive field by the corresponding kernel value:

$$Element - wise\ Product = \begin{bmatrix} Input(i,j) & Input(i,j+1) & Input(i,j+2) \\ Input(i+1,j) & Input(i+1,j+1) & Input(i+1,j+2) \\ Input(i+2,j) & Input(i+2,j+1) & Input(i+2,j+2) \end{bmatrix} \quad [09]$$

Sum all the products to obtain the value at the corresponding position $(i,\ j)$ in the output feature map:

$$Output(i,j) = \sum_{m=0}^{2} \sum_{n=0}^{2} Input(i+m, j+n) \cdot Kernel(m,n) \quad [08]$$

Stride determines the step size of the kernel as it slides across the input. For a stride of $s$, the kernel moves sss pixels along the horizontal and vertical axes:

$$Output\ Size = \left[ \frac{Input\ Size - Kernel\ Size}{Stride} \right] + 1 \qquad [10]$$

Padding adds extra pixels (often zeros) around the input's borders to maintain the spatial dimensions after convolution:

$$Output\ Size\ (with\ padding) = \frac{Input\ Size + 2p - Kernel\ Size}{s} + 1 \quad [11]$$

For multi-channel inputs (e.g., RGB images or multi-layer feature maps):

- Each kernel has $c$ channels, where $c$ is the number of input channels.
- Convolution is performed across all channels:

$$Output(i,j) = \sum_{c=1}^{C} \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} Input_c\ (i+m, j+n) \cdot Kernel_c(m,n) \quad [12]$$

For multi-channel outputs:

- Multiple kernels are used, one for each output channel.

### The 2D Maxpooling (Applied in Encoder phase)

2D Max Pooling is a down-sampling operation commonly used in convolutional neural networks to reduce the spatial dimensions of feature maps while retaining the most salient features. It works by dividing the input feature map into non-overlapping or overlapping patches and selecting the maximum value within each patch. Hence, Maxpooling reduces computational cost, mitigates overfitting, and emphasizes the most important features by focusing on maximum activations.

Mathematically, the max pooling operation can be expressed as:

$$Output(i,j) = \max_{m=0} \ (to\, p-1) \max_{n=0} \ (to\, p-1)\, Input(i \cdot s + m, j \cdot s + n) \quad [13]$$

Where:

- $Input(i,j)$: Pixel value of the input feature map at position (i,j).
- $Output(i,j)$: Output feature map value at position (i,j).
- $p \times p$: Pooling window size (e.g., $2 \times 2$, $3 \times 3$).
- $s$: Stride, determining the step size for the pooling window.
- $(i \cdot s + m, j \cdot s + n)$: Coordinates of the elements within the pooling window.

The input feature map is divided into non-overlapping or overlapping $p \times p$ windows (receptive fields) based on the stride value sss. For a stride of sss, the window for the top-left corner of the output feature map at position $(i,j)$ is defined as:

$$Window_{i,j} = \begin{pmatrix} Input(i \cdot s, j \cdot s) & \cdots & Input(i \cdot s, j \cdot s + p - 1) \\ \vdots & \ddots & \vdots \\ Input(i \cdot s + p - 1, j \cdot s) & \cdots & Input(i \cdot s + p - 1, j \cdot s + p - 1) \end{pmatrix} \ [14]$$

For each window, the maximum value is selected:

$$Output(i,j) = max\left(Window_{i,j}\right) \quad [15]$$

If $s < p$, the windows overlap, and some regions contribute to multiple output values. Conversely, for $s = p$, the pooling windows are non-overlapping.

The dimensions of the output feature map are determined by:

$$Output\ Width = \left\lfloor \frac{Input\ Width - p}{s} \right\rfloor + 1 \quad [15]$$

$$Output\ Height = \left\lfloor \frac{Input\ Height - p}{s} \right\rfloor + 1 \quad [15]$$

If padding is applied, the formulas adjust to include the padded dimensions.

- Multi-Channel Inputs:

For multi-channel inputs, max pooling is applied independently to each channel. If the input has $C$ channels, the output will also have $C$ channels, with each channel processed separately.

**The 2D Transposed Convolutions (Applied in Decoder phase)**

2D Transposed Convolutions, also known as fractionally strided convolutions or deconvolutions, are operations used to upsample feature maps. Unlike standard convolutions, which reduce spatial dimensions, transposed convolutions expand the spatial dimensions of feature maps. Hence, transposed convolutions are crucial in tasks requiring spatial dimension restoration, such as image segmentation and generation. In medical imaging, they are vital for reconstructing high-resolution segmentation maps from low-resolution encoded features, as in U-Net models.

Mathematically, a 2D transposed convolution can be represented as:

$$Output(i,j) = Input(\sum_{m=0}^{k-1}\sum_{n=0}^{k-1}\left\lfloor\frac{i+m-P}{s}\right\rfloor, \left\lfloor\frac{j+n-P}{s}\right\rfloor) \cdot Kernel(m,n) \quad [16]$$

Where:

- $Input(x,y)$: Pixel value of the input feature map at position (x, y).
- $Kernel(m,n)$: Weight of the kernel at position (m,n).
- $Output(i,j)$: Output feature map value at position (i, j).
- $k \times k$: Size of the kernel.
- $P$: Padding applied to the input feature map.
- $s$: Stride, determining the spacing between kernel applications.

The input feature map is expanded by inserting zeros between elements, depending on the stride $\bar{s}$. For example, if the stride is 2, each element in the original feature map is separated by a single zero, effectively doubling the spatial dimensions.

The kernel, with size $k \times k$, slides over the upsampled input map. At each position, the kernel performs an element-wise multiplication with the overlapping region of the upsampled input map and sums the results. Mathematically:

$$Output(i,j) = \sum_{m=0}^{k-1}\sum_{n=0}^{k-1} Upsampled\ Input(i+m, j+n) \cdot Kernel(m,n) \quad [17]$$

For example, if the kernel is:

$$Kernel = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

The result for a given sliding window is computed as:

$$Result = (1 \cdot 1) + (0 \cdot 0) + (0 \cdot 0) + (-1 \cdot 2) = 1 - 2 = -1$$

Because of the upsampling and kernel size, overlapping regions in the output map may occur. The overlapping contributions are summed during the convolution operation to form the final output.

The dimensions of the output feature map are determined by:

$$Output\ Width = s \cdot (Input\ Width - 1) + k - 2 \cdot P \qquad [13]$$

$$Output\ Height = s \cdot (Input\ Height - 1) + k - 2 \cdot P \qquad [13]$$

This formula accounts for the stride, kernel size, and padding.

For multi-channel inputs, transposed convolutions are applied separately to each channel, and the results are summed across all channels to produce the output.

# 3. Experiments

## 3.1. Experimental setup

### 3.1.1. Dataset

This dataset includes medical images obtained through breast ultrasound scans, which are essential for early cancer detection and diagnosis. The dataset contains ultrasound images categorized into three distinct classes: **normal**, **benign**, and **malignant**. These images have shown great potential for use in machine learning applications such as classification, detection, and segmentation of breast cancer.

The dataset was collected in 2018, focusing on women aged between 25 and 75 years. It includes medical images from 600 female patients. The dataset consists of 780 breast ultrasound images, each with an average resolution of 500x500 pixels and saved in PNG format. Alongside the images, corresponding ground truth labels are provided, enabling accurate model training. The images are classified into three categories: normal, benign, and malignant, providing a solid foundation for various machine learning tasks aimed at detecting and classifying breast cancer.

### 3.1.2. Evaluation metrics

We evaluate the performance of our system based of the following criteria:

$$Accuracy = IoU = \frac{Intersection\ of\ predicted\ and\ ground\ truth}{Union\ of\ predicted\ and\ ground\ truth} = \frac{|A\ \cap\ B|}{|A\ \cup\ B|}$$

Where:

A: The predicted segmentation mask.

B: The ground truth segmentation mask.

$\cap$: The intersection (common area between the two masks).

$\cup$: The union (total area covered by both masks).

$|A\ \cap\ B|$: The number of pixels that are common in both the predicted and ground truth masks.

$|A \cup B|$: The number of pixels covered by either the predicted or ground truth masks.

$$Binary\ Cross - Entropy\ Loss = -\frac{1}{N}\sum_{i=1}^{N} [y_i\ log(p_i) + (1 - y_i)\ log(1 - p_i)]$$

Where:

N: The total number of pixels (or data points) in the batch.

$y_i$: The ground truth label for pixel i, which is either 0 (background) or 1 (foreground).

$p_i$: The predicted probability for pixel i that it belongs to the foreground class (between 0 and 1).

Log: The natural logarithm.

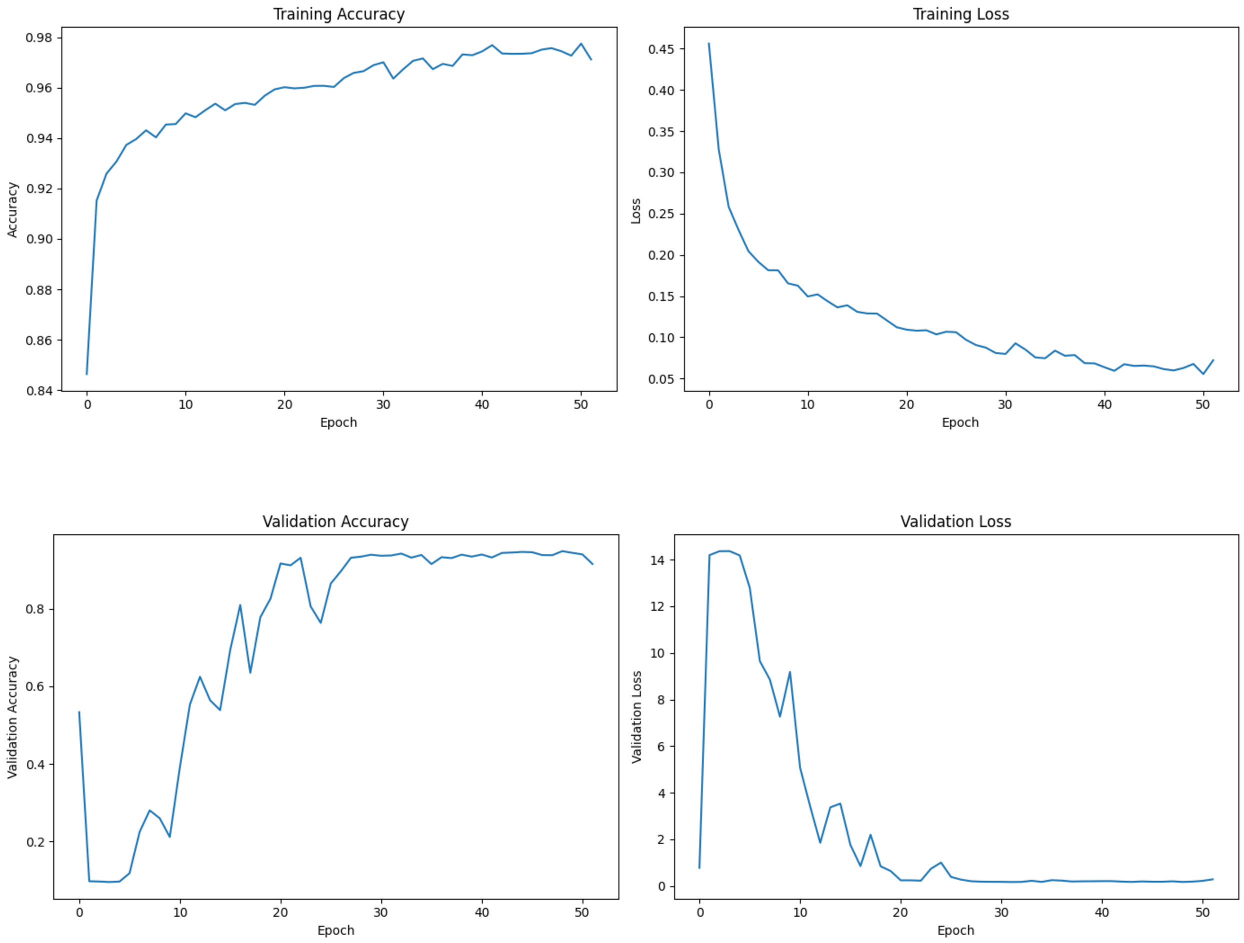
## 3.2. Experimental results

### 3.2.1. Performance of baseline system

The model performs the segmentation of areas showing signs of cancer in the images

| Set | Accuracy | Loss |
|---|---|---|
| Training | 0.9719 | 0.0726 |
| Testing | 0.9451 | 0.1680 |

.



The experimental results demonstrate that the model performs well in segmenting areas showing signs of cancer in the images. On the training set, the model achieves a high accuracy of 97.19% with a very low loss of 0.0726, indicating strong learning from the training data. On the testing set, the accuracy slightly drops to 94.51%, and the loss increases to 0.1680. This small generalization gap suggests that the model generalizes reasonably well to unseen data, though there is some room for improvement.

The training curves illustrate steady progress over epochs. The training accuracy consistently increases, nearing convergence, as seen in the "Training Accuracy" plot. Simultaneously, the training loss decreases sharply and stabilizes at a very low level, demonstrating effective and stable learning throughout the training process.

The validation curves provide further insights into the model's generalization capabilities. Validation accuracy shows a significant improvement during the initial epochs and stabilizes at a high level, closely aligning with training accuracy. Similarly, validation loss decreases sharply and plateaus, indicating that the model successfully avoids significant overfitting.

Overall, the results confirm the effectiveness of the model in segmenting cancerous areas in medical images. The small gap between training and testing performance highlights the model's robustness, but techniques like data augmentation or fine-tuning the hyperparameters could further enhance generalization. Additionally, analyzing misclassified or poorly segmented regions could provide opportunities to refine the model's performance.

### 3.2.2. Performance of different segmentations

| Model | Test accuracy | Test loss |
|---|---|---|
| SAM | 0.91 | 0.0668 |
| ResNet | 0.6923 | 1.3091 |
| U-net (Our) | 09425 | 0.1680 |

The SAM model achieves a test accuracy of 91% with the lowest test loss of 0.0668, indicating it performs well in terms of precision and error minimization. ResNet, on the other hand, exhibits a significantly lower test accuracy of 69.23% and the highest test loss of 1.3091, suggesting suboptimal performance for this task. The U-net model, which is the proposed approach in this work, achieves a high test accuracy of 94.25% and a test loss of 0.1680. While its loss is slightly higher than that of SAM, it outperforms SAM in terms of accuracy, demonstrating superior segmentation performance.

It should also be noted that the accuracy of models may vary depending on several factors beyond the model architecture itself. Characteristics of the dataset, such as being well-structured or having clear decision boundaries between classes, can significantly influence the evaluation results. Additionally, techniques like data augmentation and careful preprocessing can improve the model's generalization and robustness. These factors highlight the importance of not only choosing the right model but also ensuring the dataset and preprocessing steps are optimized for the task.

Overall, the U-net model shows a balanced trade-off between accuracy and loss, making it the most effective for the specific segmentation task. These results indicate that U-net captures complex patterns in the data better than ResNet and SAM, achieving the best overall segmentation quality. Further optimization of U-net could potentially reduce the loss further while maintaining high accuracy.

# 4. Conclusion

In conclusion, the application of U-Net in breast cancer detection represents a significant advancement in medical imaging and diagnostic accuracy. By leveraging its deep learning architecture, U-Net has shown remarkable capabilities in segmenting and identifying regions of interest in mammograms, ultrasounds, and MRIs, ultimately aiding clinicians in early cancer detection. Its efficiency in handling complex image data, coupled with its ability to generalize well across different imaging modalities, enhances diagnostic confidence and speeds up clinical decision-making.

As research and technology continue to evolve, the integration of U-Net into clinical workflows has the potential to improve patient outcomes, reduce diagnostic errors, and ensure more precise treatment planning. The future of breast cancer detection is undoubtedly intertwined with AI-driven methods like U-Net, marking a step forward in personalized and effective healthcare.

The code is published on: https://github.com/KhueHung/Breast-cancer/blob/main/v3-breast-cancer-image-segmentation-unet.ipynb

# References

[01] Yi-Sheng Sun, et al. (2017). Risk Factors and Preventions of Breast Cancer. International journal of biological sciences. doi:10.7150/ijbs.21635

[02] Stewart BW, Wild CP. (2014). World Cancer Report 2014. Geneva, Switzerland: WHO Press

[03] WHO: Geneva, Switzerland. Breast cancer. http://www.who.int/cancer/prevention/diagnosis-screening/breast-cancer/en/

[04] Lulu Wang. (2017). Early Diagnosis of Breast Cancer. Sensors. https://doi.org/10.3390/s17071572

[05] Carol E. DeSantis MPH. (2015). Breast cancer statistics, 2015: Convergence of incidence rates between black and white women. CA: A Cancer Journal for Clinicians, 66(1), 31-42. https://doi.org/10.3322/caac.21320

[06] Al-Dhabyani W, Gomaa M, Khaled H, Fahmy A. Dataset of breast ultrasound images. Data in Brief. 2020 Feb;28:104863. DOI: 10.1016/j.dib.2019.104863.

[07] Nahian Siddique, et al. (2021). U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications. IEEE Access, vol. 9, pp. 82031-82057, 2021, doi: 10.1109/ACCESS.2021.3086020.

[08] *Deep Learning* by Ian Goodfellow et al. – Chapter 9

[09] *Matrix Computations* by Gene H. Golub and Charles F. Van Loan

[10] *Neural Networks and Deep Learning* by Michael Nielsen:

[11] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

[12] *Pattern Recognition and Machine Learning* by Christopher Bishop

[13] *Deep Learning* by Ian Goodfellow and Yoshua Bengio

[14] K. He, X. Zhang, S. Ren and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904-1916, 1 Sept. 2015, doi: 10.1109/TPAMI.2015.2389824.

[15] *Neural Networks and Deep Learning* by Michael Nielsen

[16] Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science(), vol 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28

[17] Vincent Dumoulin, Francesco Visin (2016). *A Guide to Convolution Arithmetic for Deep Learning*. arXiv. https://doi.org/10.48550/arXiv.1603.07285