

# Breast Cancer Segmentation using NN-UNet

A Project Report

*Submitted by:*

**Dev Bhalodia (AU2140208)**

in partial fulfillment for the award of the degree

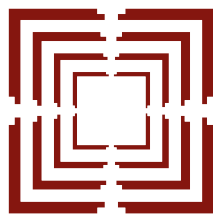
of

**BACHELOR OF TECHNOLOGY**

in

**(Computer Science and Engineering)**

at



**Ahmedabad**  
University

School of Engineering and Applied Science (SEAS)  
Ahmedabad, Gujarat

May, 2025

## DECLARATION

I hereby declare that the project entitled “**Breast Cancer Segmentation using NN-UNet**” submitted for the B. Tech. (**Computer Science and Engineering**) degree is my original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

**Signature of Student**

**Date: 24th April, 2025**

**Place: Ahmedabad**

## CERTIFICATE

This is to certify that the project titled “**Breast Cancer Segmentation using NN-UNet**” is the bona fide work carried out by **Dev Bhalodia**, a student of B. Tech. (**Computer Science and Engineering**) of School of Engineering and Applied Science at Ahmedabad University during the academic year 2023-2024, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in **Computer Science and Engineering** and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

This project was done under the supervision of the faculty mentor **Professor Jayendra Bhalodiya**.

**Signature of Faculty Mentor**

**Date: 24th April, 2025**

**Place: Ahmedabad**

## Abstract

Breast cancer has been a life-threatening disease that is most prevalent among women worldwide. It has raised global concern in the field of medical science. Robust tools that facilitate the diagnosis and detection of cancer have been a necessity. Dynamic contrast-enhanced magnetic resonance imaging (DCE-MRI) is a relatively new technique that has been adopted worldwide for its ability to provide a high-resolution image of tumor shape and enhancement pattern. In this project, I have used the self-configuring nnU-net architecture for automatic 3D breast cancer segmentation. The model is trained using a publicly available dataset, BreastDM, containing DCE-MRI scans, which are manually annotated by a team of radiologists. The nnU-net is a self-configuring framework, requiring no user intervention during preprocessing and training the volumetric data. The model achieved a Dice similarity coefficient (DSC) of 0.87, thereby surpassing the established results obtained by models used in the reference paper, such as 3d Unet (0.665), 3D-DenseSeg (0.67) and 3DVnet (0.656). Hence, from the results, we can see that the model effectively handles complex tumor structures.

# Table of Contents

Declaration . . . . .	i
Certificate . . . . .	ii
Abstract . . . . .	iii
Table of Contents . . . . .	iv
<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>vii</b>
<b>Gantt Chart</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Project Definition . . . . .	1
1.2 Project Objectives . . . . .	1
<b>2 Literature Survey</b>	<b>3</b>
2.1 Related Work . . . . .	3
2.2 Tools and Technologies . . . . .	3
<b>3 Methodology</b>	<b>5</b>
3.1 Dataset Description . . . . .	5
3.2 Data Preparation . . . . .	6
3.3 Data Preprocessing in pipeline . . . . .	7
3.4 nnU-net Architecture . . . . .	9
3.4.1 3D full resolution architecture: . . . . .	10
3.5 Training pipeline: . . . . .	11
3.6 Inference . . . . .	13
3.7 Post-processing: . . . . .	13
3.8 Visualization: . . . . .	13
3.9 Evaluation: . . . . .	14
<b>4 Results</b>	<b>15</b>
4.1 Project Outcomes . . . . .	15
4.1.1 Evaluation metrics result . . . . .	15
Default metrics result . . . . .	15
Metrics to be used . . . . .	15
Results Before Post-Processing . . . . .	16
Results After Post-Processing . . . . .	16
Visualization . . . . .	16
4.1.2 Interpretation . . . . .	18

4.1.3	Initial works . . . . .	18
	Using 2D ultrasound dataset . . . . .	18
	Transition to BreastDM 2D Dataset . . . . .	19
	Implementing Vanilla 3D U-net on 3D dataset . . . . .	20
4.2	My Contributions to the Project . . . . .	20
4.3	Learning Outcomes . . . . .	21
4.4	Real World Applications . . . . .	22
<b>5</b>	<b>Conclusion</b>	<b>23</b>
5.1	Future expansion . . . . .	24
	<b>Bibliography</b>	<b>24</b>

# List of Figures

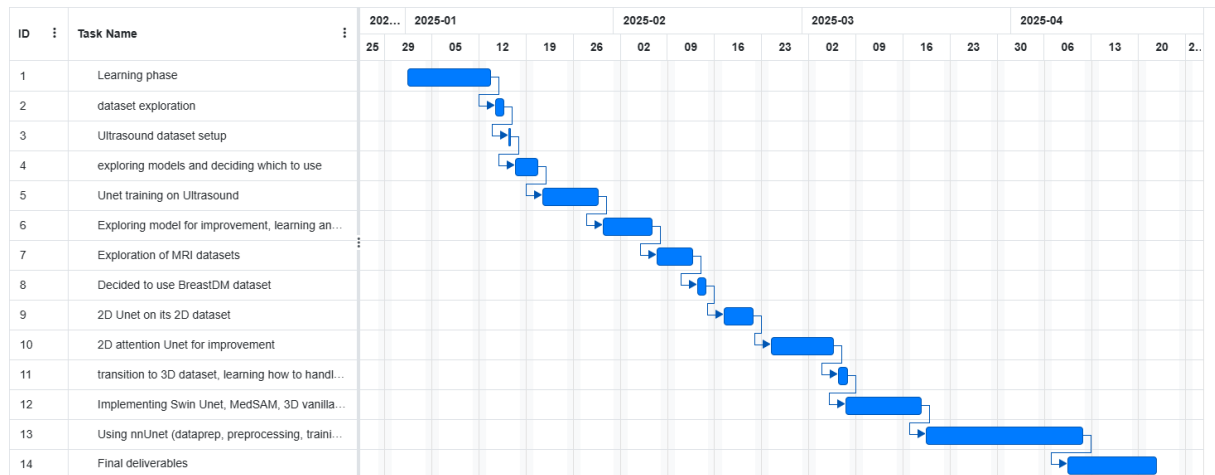
3.1.1 Visualization of input modalities. . . . .	5
3.4.1 Network pipeline . . . . .	9
3.4.2 3D full resolution configuration . . . . .	9
4.1.1 Progress graphs showing trends over the epochs. . . . .	17
4.1.2 Visual comparison of segmentation results. . . . .	18
4.1.3 Segmentation output (using Attention U-net) . . . . .	19
4.1.4 Segmentation output (using Attention U-net) . . . . .	20

# List of Tables

4.1.1 Baseline vs. Proposed Results on 3D BreastDM Dataset . . . . .	16
4.1.2 Baseline vs. Attention U-net Results on 2D BreastDM Dataset . . . . .	19



# Gantt Chart



# Chapter 1

## Introduction

From all the cancers, Breast Cancer maintains the stand of being the most commonly diagnosed cancer among women. It has been a primary cause among cancer related deaths. Early diagnosis, that too, accurate, is significant in order to treat the patient rightfully and prevent death. **MRI, Dynamic Contrast Enhanced** in particular, has been most commonly used modality for its superior soft tissue contrast and high sensitivity. This technique has efficiently helped in detecting breast tumors effectively.

Because manually annotating the tumors has been time-consuming and requires experts, deep learning techniques have been used for automating the process. Not only automation, different models are built and customized for different biomedical image segmentation tasks.

### 1.1 | Project Definition

The main goal of this project is to use a fully automated pipeline for 3d segmentation using **nnU-Net**. It is a self configuring deep learning framework which has been used for preprocessing the **BreastDM** dataset, its training and evaluation. The dataset comprises of DCE-MRI scans along with expert annotations.

The nnU-net automatically adapts to the dataset, unlike other conventional models, which require manual tuning and implementation of the architecture. The main goal is to evaluate its performance and compare it against results of other models trained under this same dataset.

### 1.2 | Project Objectives

The principal objectives of this project are:

- To perform initial stage of preprocessing by converting the BreastDM dataset into a format which is compatible with the framework.
- On these compatible MRI scans, applying 3d full resolution pipeline of **nnU-net** for tumor segmentation.
- To evaluate the model's performance using metrics such as Dice Score and compare it with 3D models like 3D Unet, 3D VNet, and 3D-DenseSeg which have been used on the same dataset.

- To perform qualitative analysis through visualization and postprocessing to assess the model's clinical viability.

# Chapter 2

## Literature Survey

Breast cancer segmentation has been a preliminary step where the MRI scans are analyzed to find the tumors. Accurate segmentation of tumors is necessary for correct diagnosis and planning of treatments. There have been various advancements in deep learning, specifically in CNNs, attention-based models and transformers where many techniques have been proposed to automate the segmentation process. Although there have been many improvements over the years, the proposed models have faced several challenges in handling complexities like small and irregular tumors and noise. Below are some existing works in this area.

### 2.1 | Related Work

These are some of the previously used models for generating segmentation from MRI images:

- **U-Net (2D/3D):** A fundamental architecture for medical image segmentation. It is effective for handling 2d images, but struggles with 3d images due to complex spatial relationships within slices.
- **Attention U-Net:** Adds attention gates to the basic U-net, thereby strengthening the capabilities of U-net. It handles the small irregular tumors better than U-net, but adds computational load.
- **3D VNet:** It is designed to handle 3D data in particular. It still faces difficulties while dealing with small irregular tumors.
- **3D DenseSeg:** It has dense connections within layers, and it promotes feature reuse. It performs good on limited data.
- **Transformer-based Models (e.g., Swin UNet, MedSAM):** Swin UNet is a vision transformer based model which captures long-range dependencies. The MedSAM is a pre-trained transformer model. These models are used only when there are large datasets, making them computationally expensive.

### 2.2 | Tools and Technologies

The primary tools and technologies used in this project are as follows:

- **Programming Language:** All the models that I have implemented use Python as the primary language.
- **Deep Learning Frameworks:**
  - **PyTorch:** It is a deep learning framework known for its scalability and flexibility.
  - **nnU-Net:** Automated pipeline used for its robustness in segmentation tasks.
- **Libraries:**
  - **NumPy:** Plays a role in handling large multi-dimensional arrays and matrix operations.
  - **Matplotlib:** Has been used for plotting graphs and visualizing images.
  - **Nibabel:** For read and write operations of NiFTI files.
  - **OpenCV:** for reading, writing, and manipulating images and videos.
- **Hardware & Software:**
  - **CUDA:** As Tesla T4 GPU is required for training, CUDA is used for GPU acceleration.
  - **Google Colab:** As it provided free GPU and a space to smoothly implement the model, Colab is used.

# Chapter 3

## Methodology

### 3.1 | Dataset Description

#### Source of Data

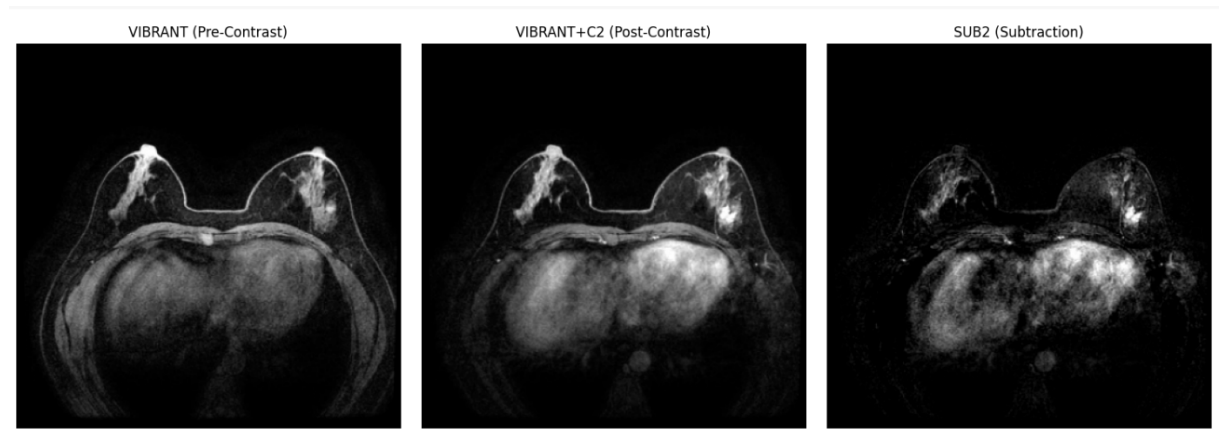
**BreastDM** is a publicly available dataset, which is used in this project. It comprises of DCE-MRI scans of 232 patients, out of which 147 are malignant cases and 85 benign cases.

#### Characteristics of Dataset

Each patient data has three 3D MRI modalities saved in `.npy` format:

- **VIBRANT.npy** – pre-contrast image
- **VIBRANT+C2.npy** – post-contrast image
- **SUB2.npy** – subtraction image (post-contrast minus pre-contrast)

Each modality has 8 number of slices of dimension 369X369. In case of images, all the 3 modalities are used. In case of labels, only the **SUB2.npy** modality is used as a ground truth mask.



**Figure 3.1.1:** Visualization of input modalities.

## 3.2 | Data Preparation

### ■ Dataset Structure Planning:

- Each Patient data contains 3 modalities: `VIBRANT.npy`, `VIBRANT+C2.npy`, and `SUB2.npy`.
- In case of image data, all the 3 modalities were to be used.
- Only the `SUB2.npy` modality was used as ground truth masks.

### ■ Removal of Blank Slices:

- Each modality consists 8 slices. There were many patient samples in the original dataset where slices were padded to 8, as in, blank slices were added to get a volume of 8 slices. I removed these blank slices from all the 3 modalities of image data and corresponding label while still in the `.npy` format.

### ■ Conversion to `.nii.gz` Format:

- using `nibabel`, all the `.npy` files were converted into `.nii.gz` format, as it is one of the many compatible formats with nnU-net.
- The data was then structured according to the directory rules provided by the nnU-net documentation:
  - `imagesTr/`: training images in `.nii.gz` format.
  - `labelsTr/`: corresponding labels in `.nii.gz` format.
  - `imagesTs/`: test images in `.nii.gz` format.
- nnU-net also has file naming conventions which were to be followed:
  - Images: `BC_***.****.nii.gz` where `***` suggests the number of image we are on and `****` suggests the number of modality we are on, both starting from index 0.
  - Labels: `BC_***.nii.gz` where `***` suggests the label we are on. As only 1 modality is required for the label, we have not kept `****`.
  - Example: for image, say, we have `BC_001_0002.nii.gz` which means we are on image 2 and 3rd modality. for label, `BC.001.nii.gz` is the corresponding ground truth mask.

### ■ Label Remapping:

- Before preprocessing, the label values were remapped from 255 to 1 for the foreground, to conform with nnU-net's label format requirement.

### ■ Defining `dataset.json`:

- channel (modality) names.
- label classes (in my cases, it's just 2 classes because I am doing binary segmentation).
- Number of patient samples for training.
- file ending (`.nii.gz`).

Even the `dataset.json` is validated in order to make sure it is compatible with pipeline.

### 3.3 | Data Preprocessing in pipeline

#### ■ Cropping:

- Every volume is selected and cropped keeping the non zero regions and eliminating blank spaces.
- Size of the volume gets reduced, lowering the computational cost.
- Fingerprint is created which stores cropped image size, spacing, modalities, number of classes and number of training samples.

#### ■ Resampling (Voxel Spacing Standardization):

- Samples have heterogeneous voxel spacing.
- Median spacing value is calculated to be the target spacing for all the voxels here:
  - Images are interpolated using 3rd-order spline interpolation. The new voxel spacing is the median value of all voxel spacing in entire dataset. The intensity of voxel at new position is calculated on the basis of neighboring voxels, where a 3rd degree polynomial is made to pass through these values and new values are plugged in to estimate the intensity.
  - Segmentation masks are interpolated using nearest-neighbor interpolation where in value of the nearest voxel in the original label is copied to the new position of that voxel after resizing.
- This is done to standardize the spatial resolution, making the learning consistent and preventing artifacts (errors which aren't part of the original image before resizing).

#### ■ Configuration selection:

##### □ 2D U-Net:

- Each 2D slice of the volume is taken as an independent unit, thereby not considering the spatial information on the z axis.
- Because it is processing just 2D slices individually, less computational cost and faster will be the training.
- If the dataset consists of important 3D relationship, this configuration is not advisable due to loss of that information.

##### □ 3D U-Net (Full Resolution):

- It takes into account, the relationships across all the 3 axis, and hence, entire 3D image is processed here in full resolution leading to accurate segmentation of even complex structures.
- The only con here is that it gives a high computational load.
- It should be used in datasets where there is significant spatial information along the z-axis, like 3D MRI in my case.

##### □ U-Net Cascade:

- It has 2 stages. The 1st stage includes processing downsampled images to get initial segmentation, and use this output as an input for 2nd stage where segmentation is refined at full resolution.



- More computationally expensive due to the two-stage process.
- It should be used when you have large datasets, so that in the 1st stage, memory usage gets limited.

#### ■ Why 3D Full Resolution Was Chosen in my Case:

- As I am using 3D MRI data, it has crucial information along the z-axis, hence processing 3D volume as a whole allows the configuration to learn these spatial relationships.
- If 2D Unet were to be used, it would have lost this information during training.
- Though 3D full res Unet requires more memory and computational effort, it is the only configuration suitable for my dataset. Hence, during the preprocessing, it was this configuration which was selected according to my dataset.

#### ■ Z-score Normalization:

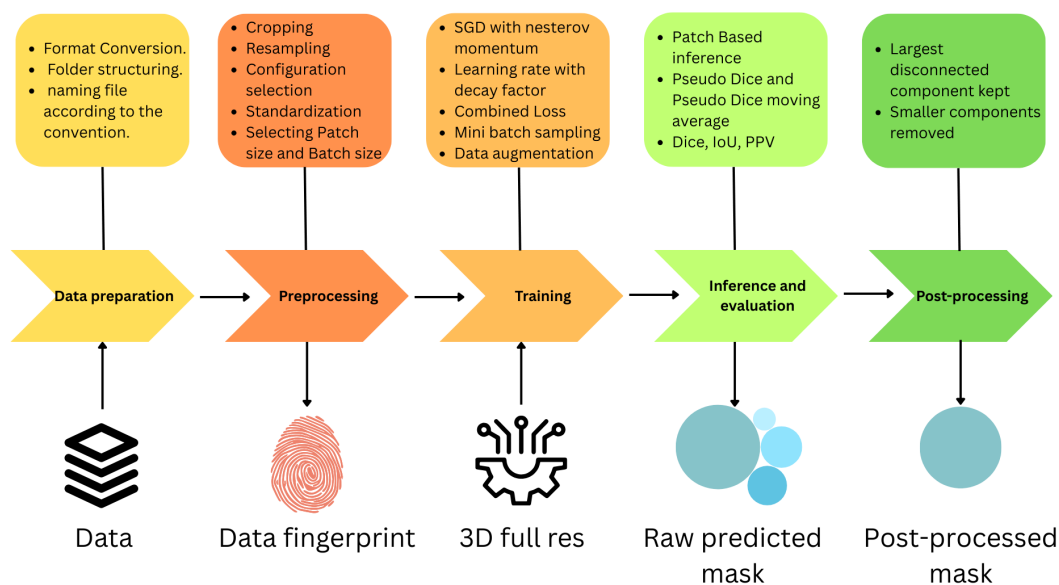
- **Z-score normalization** is applied on each 3D volume as each patient sample has varying intensities:
  - This is done so that the model doesn't care how light or dark the scans are and focuses only on learning the patterns and structures. Below is the formula for the same:

$$x' = \frac{x - \mu}{\sigma}$$

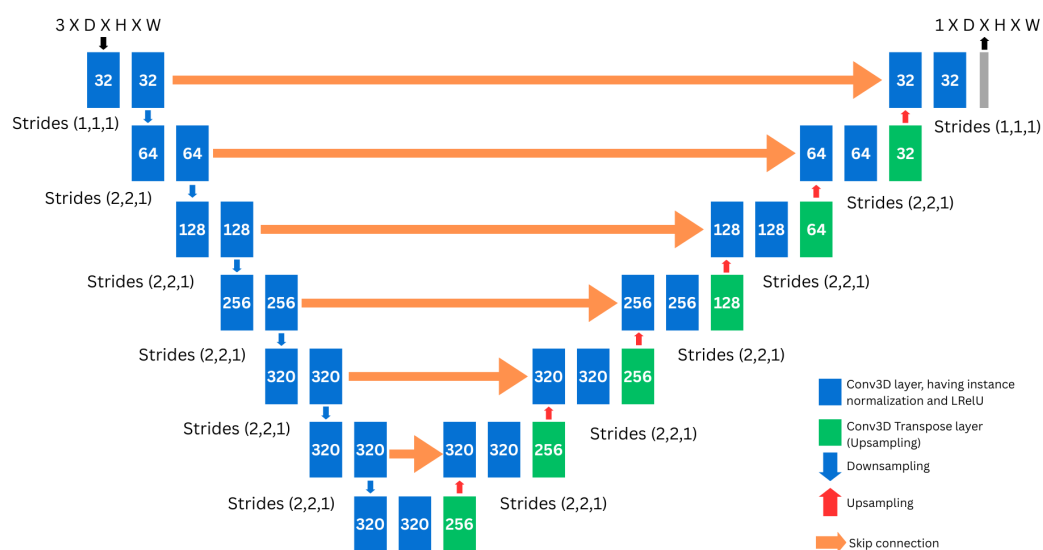
where:

- $x$  is original intensity
  - $\mu$  is mean intensity calculated considering non-zero voxels in the volume
  - $\sigma$  is standard deviation of the non-zero voxels in volume
- Conditional Z-score normalization also takes place if cropping reduces the volume size of a patient by more than 25%. Under such conditions, z-score normalization takes place only within the foreground (non zero) region whereas voxel intensities outside this region are set to 0.

### 3.4 | nnU-net Architecture



**Figure 3.4.1: Network pipeline**



**Figure 3.4.2:** 3D full resolution configuration

### 3.4.1 | 3D full resolution architecture:

The **nnU-net framework** had selected **3D full resolution U-net** according to my dataset. Even though this configuration is built using the basic principles of vanilla 3D Unet, it is dynamic and introduces changes according to the dataset being fed to it, unlike vanilla 3D Unet, where everything is fixed.

#### Differences Between Vanilla 3D U-Net and 3D Full-Resolution U-Net:

##### ■ Dynamic Dataset-Specific Configuration:

- According to the dataset given to it, it dynamically modifies the input patch size, number of pooling operations, width and depth, and also batch size considering the memory of GPU.
- Everything in vanilla Unet is predetermined, and hence fixed. So, it will not generalize well for each and every dataset given to it.

##### ■ Preservation of Full Spatial Resolution:

- As the name suggests, it maintains full resolution. For that, it will need to minimize aggressive downsampling in order to keep crucial spatial information. Based on the median shape and voxel spacing of the data, it will select the number of pooling operations and strides carefully.
- Due to significant retaining of spatial information, it will be able to detect complex small irregular tumors as well.

##### ■ Patch-Based Training:

- Dynamically defines patch size according to data's median shape and spacing. These patches will ensure that same amount of context is seen by the model, across various patients.
- Patch size in Vanilla Unet is not according to the dataset given to it. It is rather arbitrarily generated.

**Encoder-Decoder Backbone with Skip Connections** The nnU-Net full-res model is built using the basic structure of vanilla Unet:

- **Encoder:** Semantic features are extracted here using pooling layers and convolution operation.
- **Decoder:** The compressed features from the encoder are upsampled here, thereby rebuilding the spatial resolution.
- **Skip Connections:** Feature map from Encoder (before going through convolution) is fed to the corresponding decoder layer, for retrieving important spatial information lost during downsampling.

## Architectural alterations

- **Activation Function:** Instead of RelU, which caused dying RelU problem in case of negative inputs, **Leaky RelU** is used. In case of negative input, it converts to small positive value (0.01).
- **Normalization:** Because small batches are used for limiting the memory usage and there is varied intensity distribution in the images, **instance normalization** is used instead of batch normalization. Here, each sample (patch) of the batch is normalized individually, where as in case of batch normalization, a whole batch is used.
- **Patch-Based Inference:** The nnU-net does not feed the entire volume into the model but small overlapping patches. After predictions are done, the predicted outputs are placed back in their original position inside volume. It uses Gaussian weighting average where voxels in the center are given more than those near borders.

## Why is this Configuration suitable for my data?

- As the full resolution is tried to maintained, there is reduced loss of the spatial information and details are hence preserved.
- As this configuration adapts to various datasets, it has proven to generalize well when my dataset was fed to it.

## 3.5 | Training pipeline:

### ■ Training Schedule:

- The training continues till 1000 epochs or till convergence. In my case, I did not see any improvement after 38th epoch and hence trained till 53 epochs.
- **250 mini batches** are trained in 1 epoch.

### ■ Optimizer:

- **Stochastic Gradient Descent (SGD)** with **Nesterov momentum** is used.
- The Momentum coefficient is  $\mu = 0.99$ , which means that 99% of the previous update's direction is carried forward.
- The learning rate initially is set to 0.01.

### ■ Learning Rate Decay:

- This is how the learning rate is updated:

$$lr = lr_{\text{init}} \times \left(1 - \frac{\text{epoch}}{\text{epoch}_{\text{max}}}\right)^{0.9}$$

- At first, the updates are bigger due to greater lr. With every epoch, lr decreases, allowing fine-tuning of the weights in order to learn better. Also, the decrease is gradual, preventing the model from getting stuck in a local minimum.

### ■ Loss Function:

- Combined loss is computed, comprising of **Dice loss** and **Cross Entropy loss**.
- Total loss:

$$L_t = L_{\text{dice}} + L_{\text{ce}}$$

- Dice loss:

$$\text{Dice Loss} = 1 - \frac{2 \cdot |A \cap B|}{|A| + |B|}$$

where

$A$  = predicted mask,

$B$  = ground truth mask,

$|A|$  = predicted mask volume,

$|B|$  = ground truth mask volume

- Cross entropy:

$$\text{CE Loss} = - \sum_i y_i \cdot \log(p_i)$$

where

$y_i$  = the ground truth value for voxel  $i$  (*either 0 or 1*),

$p_i$  = the predicted probability for voxel  $i$ .

#### ■ Mini-Batch Sampling Strategy:

- From the dataset, there is random selection of samples.
  - 66.7% of patches: Samples that are selected from random places.
  - 33.3% of patches: will definitely have non zero region, demonstrating the class (in my case, it is 1 for foreground)

#### ■ Data Augmentation (on-the-fly):

- The model requires a substantial amount of data in order to prevent overfitting and to generalize well.
- Augmentation techniques include:
  - Rotations
  - Scaling
  - Gaussian noise addition
  - Gaussian blur
  - Brightness and contrast adjustments
  - Low resolution simulation
  - Gamma correction
  - Mirroring

## 3.6 | Inference

### ■ Patch based inference:

- As discussed earlier, during the inference, a whole volume is not taken but its overlapping patches are taken for prediction. This will limit the memory usage.
- The whole image volume is split into overlapping patches which go through inference patch-wise. The predictions are then merged using gaussian weighting in order to form fully segmented volume.

### ■ Gaussian weighting:

- Gaussian weighting average scheme is used here for combining predictions from overlapping patches.
- In order to merge the predictions smoothly, it gives higher weight to the center voxels and lower weight to the border voxels. This is done as voxels at the center hold significant information and voxels at boundaries don't hold much information and to prevent from border artifacts, thus eliminating the boundaries between the merged predictions.

## 3.7 | Post-processing:

- The outputs of inference, that is, the predicted labels, go through post processing where the segmentation is refined, and the final segmentation is an improved version of itself.
- Post processing is mainly done for following reasons:
  - **Small false positives:** If in the segmentation, a tumor (being small segments) is also detected at a wrong place, being false positive or likely just noise, they are removed.
  - **Disconnected components:** It is assumed that tumors are contiguous and hence only the largest connected component is kept.
- In the nnU-net pipeline, the post processing happens only after completing whole training of 1000 epochs. As I trained only for 53 epochs, I had to write a python script for manual post processing.

## 3.8 | Visualization:

- Visualization enables qualitative assessment of the model's performance.
- Visualization is done for side by side comparison, where in all the 3 modalities of image, ground truth mask, predicted mask and post processed mask, are visualized side by side to have a clear view.

### 3.9 | Evaluation:

- In the pipeline, 2 metrics are evaluated. Pseudo dice score which is the dice score of the entire mini-batch and Pseudo dice score moving average with a smoothing factor.
- But, as I had taken a research paper for comparing the final results, I have taken these metrics they have used for evaluating their model:
  - **Dice Score Coefficient:** Measures the overlap between ground truth mask and the predicted mask.
  - **IoU (intersection over union):** Similar to dice, but a stricter metric comparatively.
  - **Positive predicted value:** It gives us the accuracy of the predictions that are positive by calculating how many positive values that are predicted, are actually correct.

# Chapter 4

## Results

The main goal of the project was to use the recent deep learning framework in order to develop an efficient and effective breast cancer segmentation model using 3D DCE-MRI dataset. The objective was to use an architecture not much explored in the segmentation of breast cancer using BreastDM dataset, in order to see whether it is outperforming the other models (which have used the same dataset) or not. Hence, I aimed to compare performance of this model with the models used in the reference research paper.

### 4.1 | Project Outcomes

#### 4.1.1 | Evaluation metrics result

##### Default metrics result

The best model which got saved as `checkpoint_best.pth` showed the following result:

- Pseudo Dice Score: **0.8920**
- Pseudo Dice moving average: **0.8748**

##### Metrics to be used

##### Dice Score (DSC):

$$\text{DSC} = \frac{2 \times |A \cap B|}{|A| + |B|}$$

##### Intersection over Union (IoU):

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|}$$

##### Positive Predictive Value (PPV):

$$\text{PPV} = \frac{|A \cap B|}{|B|}$$

##### Where:

- $A$  = set of ground truth positive pixels (true mask)
- $B$  = set of predicted positive pixels (predicted mask)



## Results Before Post-Processing

Using the raw predicted labels, the metrics below were computed:

- Mean Dice Score: **0.840**
- Mean IoU Score: **0.744**
- Mean PPV Score: **0.799**

## Results After Post-Processing

The results have significantly improved after postprocessing. Below are the metrics computed using the labels refined after going through post processing:

- Mean Dice Score: **0.869**
- Mean IoU Score: **0.805**
- Mean PPV Score: **0.876**

A comparison with baseline models reported in the BreastDM paper is shown below:

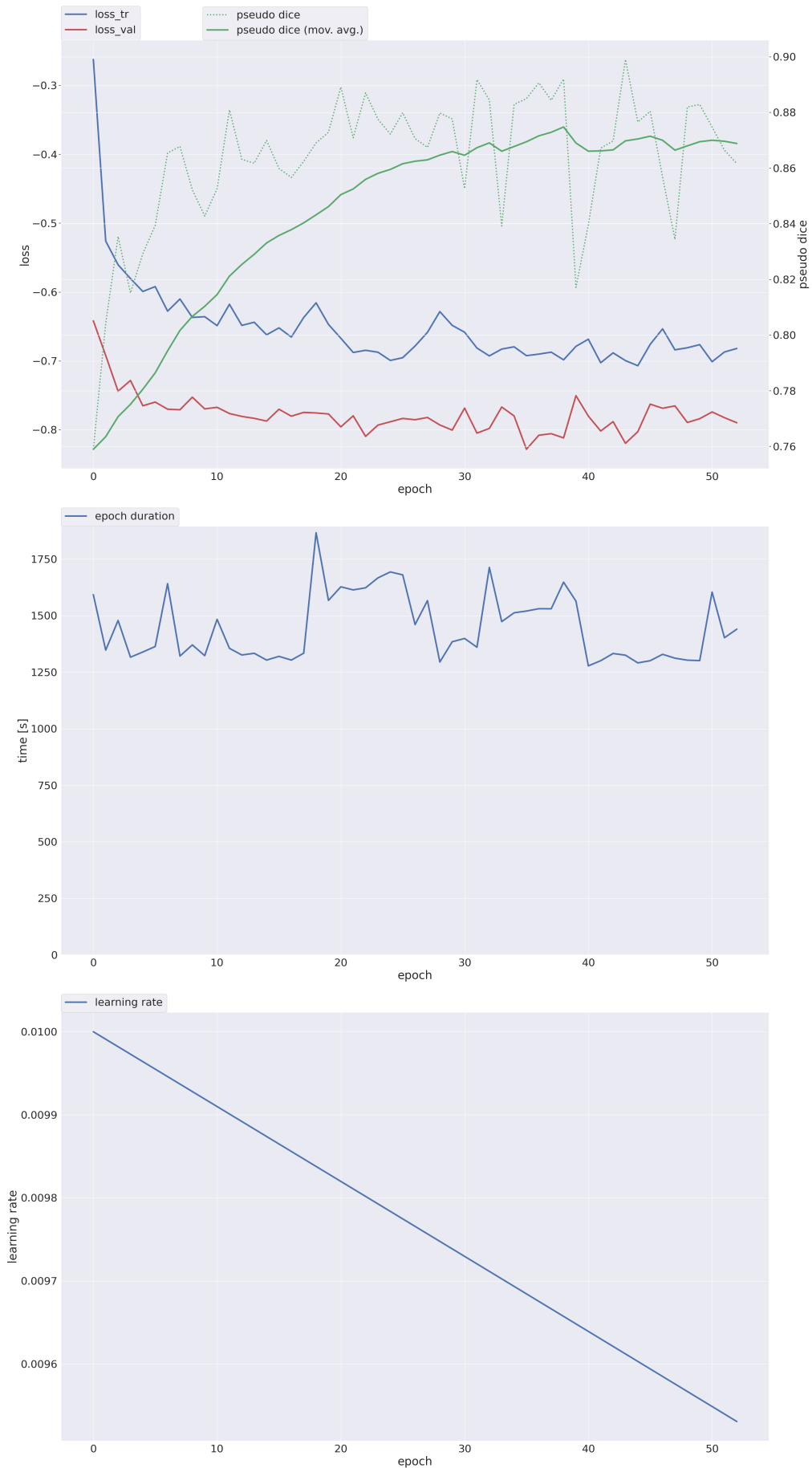
**Table 4.1.1:** Baseline vs. Proposed Results on 3D BreastDM Dataset

Model	DSC (%)	IoU (%)	PPV (%)
3D U-Net (Paper)	66.5	75.1	83.3
3D V-Net (Paper)	65.6	74.2	77.1
3D DenseSeg (Paper)	67.0	74.9	84.1
<b>3D nnU-Net</b>	<b>86.9</b>	<b>80.5</b>	<b>87.6</b>

## Visualization

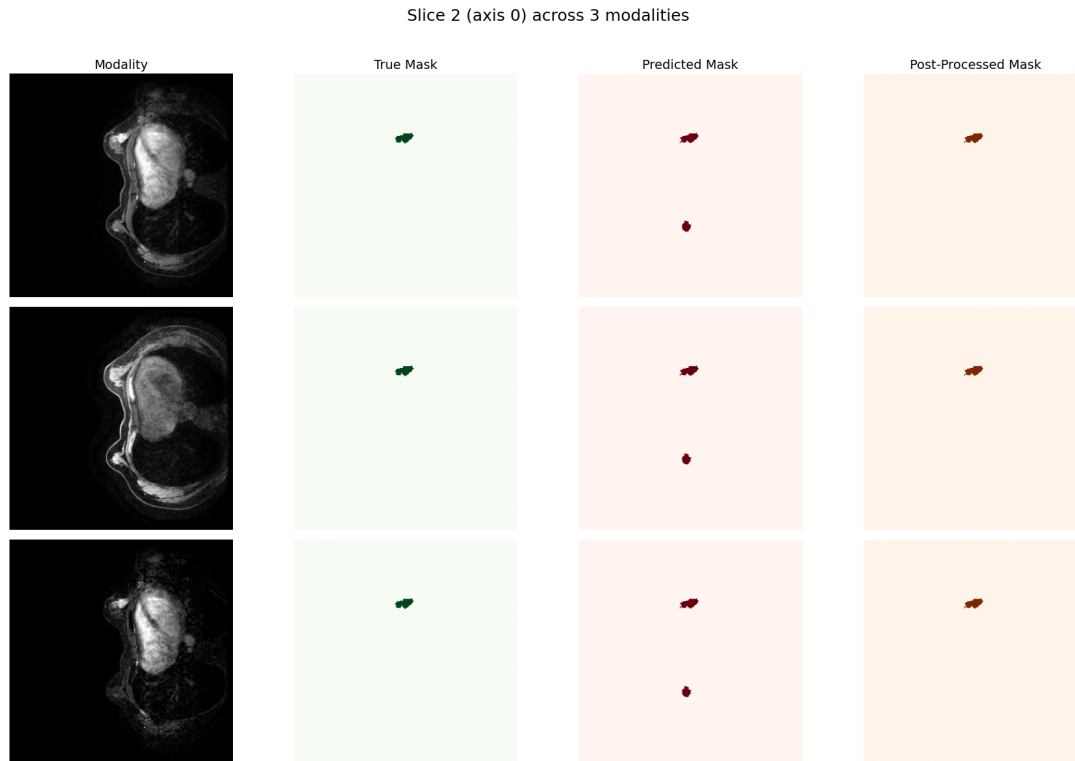
With the help of pipeline, graphs showing progress during the training was generated. Also, in order to assess the model's performance visually, segmentation output are plotted.

- The first graph below shows train loss, validation loss, pseudo dice score and pseudo dice moving average over the epochs trained. As you
- Then second graph shows the duration for which each epoch ran (25 minutes on average).
- The third graph shows the decaying of learning rate which is gradual (decreasing by 0.001 each epoch).



**Figure 4.1.1:** Progress graphs showing trends over the epochs.

**Segmentation Outputs** Figure 4.1.2 provides a side by side comparison of 4 different units. The first is the actual image (1st modality). The second is the corresponding true mask. The third is the raw predicted mask (having small disconnected component being false positive). The last is the the postprocessed label having that false positive mask removed. This was for first slice of each image. This is done for each modality and each slice of the image.



**Figure 4.1.2:** Visual comparison of segmentation results.

### 4.1.2 | Interpretation

With this model used, it clearly surpassed the results which were obtained by the baseline models in this research paper. Also, the results were further improved by postprocessing the raw predicted masks, making the output more reliable.

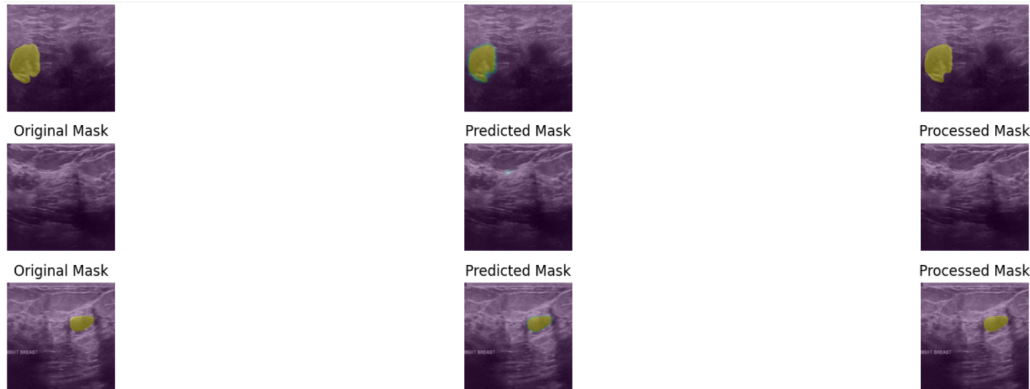
### 4.1.3 | Initial works

#### Using 2D ultrasound dataset

I started by experimenting with the 2D ultrasound breast cancer dataset provided by kaggle. I used it as it was readily available dataset and least complex directory structure. The data was almost ready-to-use with not much preprocessing required. Also, because the dataset was less complex, it gave satisfactory results with basic models.

- I implemented the most common architecture, basic U-net first. With proper hyperparameter tuning, I got **0.46** mean IoU.

- To improve the results further, I used Attention U-net which used attention gates, focussing on only relevant regions and ignoring irrelevant backgrounds. With proper hyperparameter tuning, I was able to acheive mean IoU of **0.54**.



**Figure 4.1.3:** Segmentation output (using Attention U-net)

### Transition to BreastDM 2D Dataset

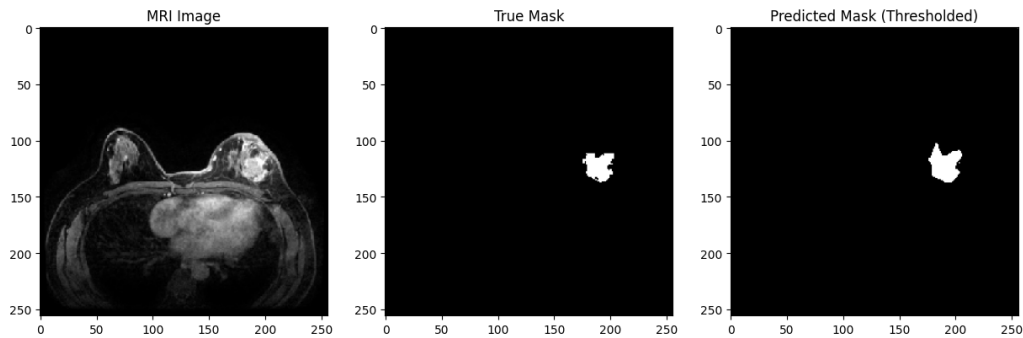
As my main problem statement was to work with MRI images, I started to find a relevant dataset. I got the BreastDM dataset, which also includes 2d images as well. As I did not have proper knowledge about dealing with 3D data, I started with the 2D data first using the same 2 models above.

- With U-Net, the model got a mean IoU of **0.49**.
- Using Attention U-net showed significant improvement as it got a mean IoU score of **0.928**.

The reference research paper also had used the 2d BreastDM dataset. Below is the comparison of the results obtained from models they used with the result I obtained by using Attention U-net:

**Table 4.1.2:** Baseline vs. Attention U-net Results on 2D BreastDM Dataset

Model	DSC (%)	IoU (%)	PPV (%)
U-Net (Paper)	73.7	80.3	83.6
U-Net-VGG16 (Paper)	70.4	77.3	79.6
UNeXt (Paper)	70.1	78.5	86.3
PSPNet (Paper)	72.7	79.0	72.9
<b>Attention U-Net</b>	<b>82.5</b>	<b>92.8</b>	<b>72.1</b>



**Figure 4.1.4:** Segmentation output (using Attention U-net)

### Implementing Vanilla 3D U-net on 3D dataset

When I transitioned to 3D dataset of the BreastDM, I started with the implementation vanilla U-net. I used all the 3 modalities of images here as well by stacking them manually creating 4d tensor (slices, channel, height, width). I also implemented on-the-fly augmentation for better generalization and did fine tuning. Getting to know that this model is already implemented by them, I did not run it for enough epochs and at the end of 18th epoch, I got dice score of **0.54**.

## 4.2 | My Contributions to the Project

- The default format was not aligned with the format compatible with the nnU-net. Hence, as you already went through the data preparation process, I wrote multiple scripts converted my data into specified format and structured it according to the nnU-net
- Trained on various datasets, dimensions, and models in order to come up with more better solutions initially.
- Explored many 3D models such as Vanilla 3D Unet and then explored models not much used in the domain of breast cancer like Swin U-net and MedSAM before coming to nnU-net.
- Learned how to run the pipeline till the end completely.
- I was not able to use the post processing provided by the pipeline as I it could be used only after complete training till 1000 epochs. Hence, I coded the post processing in order to retain only the largest disconnected component from the raw predicted masks.
- The model was evaluated on the pseudo dice score and pseudo dice moving average. These 2 metrics weren't used in the research paper. Hence, I had to manually evaluate the test data using dice score, mean IoU, and PPV as the metrics.
- Efficiently visualized the segmentation output for a clear look at the model's performance.

## 4.3 | Learning Outcomes

### **Built foundational knowledge in the field of AI/ML:**

- I did not have any prior knowledge in the field of AI/ML. I gradually learned with theory and hands on by first starting to create mini projects using deep learning techniques and then proceeding towards the project.

### **Interpretation of 2D and 3D medical images:**

- Learned about different datasets and different modalities associated to it. Learned handling different datasets through practical works.
- I had never come across 3D data before. I learned to properly handle 3D images.

### **Deep learning models:**

- Explored and learned how to implement different models, for both, 2D as well as 3D dataset.
- Learned about how they work and how they perform differently and give varying results. For improving the results, I was able to explore correct models which successfully showed improvements when compared to previous results.

### **Handling the data and preprocessing it:**

- No data is readily trainable. I had to make the data ready to train so that the training becomes error free and used memory efficiently. Moreover, I had fine tune in such a way that it gives me better results.
- I learned how to load and preprocess data in order to make it compatible to the selected model, also making sure it does not take too much time and memory to get processed during training.
- I learned how to normalize, do padding with interpolation operation, and do dimensional adjustments.

### **Training and evaluation:**

- Explored different loss functions and optimizer to compare and get the best results.
- Learned how to apply transformations on the image (rotating, flipping, shifting, scaling, etc.) in order to augment the data so that the model generalizes well.
- Explored different metrics to evaluate the model, like accuracy, dice score, mean IoU, and PPV.

### **Postprocessing and inference:**

- I learned to develop customized post processing in order to improve the segmentation output.

## Visualization

- As images and labels were of different dimensions, I learned to write customized scripts catering to the data requirements in order to visualize them. I also learned to plot the train curves which give us valuable insights of model's performance during training and whether to train it more or not.

## 4.4 | Real World Applications

- **Reducing Radiologist Workload**

The burden of manually annotating the masks is taken away from the experts. Manual annotation, being time consuming, is also subjected to errors due to variability between different doctors. It also gives the experts, more time to make decisions rather than getting engrossed in repetitive task.

- **Suitable in resource limited places**

In the places where there are limited radiologists, it could be for diagnostic purposes thus enhancing the breast cancer management in such places.

- **Integration into PACS**

This automated system can be integrated with Picture Archiving and Communication System (PACS), where in as soon as the MRI scans get stored, they are automatically segmented.

- **Research and Clinical studies**

The segmented output can be used in clinical studies for side by side comparison with the MRI scans for learning how to locate the tumor. As the model I used can be used for various dataset of different body parts, it is helpful to many researchers who can bring their data and use this model for benchmarking.

- **Classifying the tumor**

After the segmentation output (predicted and post-processed label) is obtained, it can be further used for classifying whether it's benign or malignant by computing size, shape, volume, texture, etc and then feeding it to the classifier.

# Chapter 5

## Conclusion

A strong deep learning based segmentation system was developed in this project which used volumetric breast DCE-MRI scans in order to give automated segmentation outputs. I started with no prior experience in the field of AI/ML, and my journey involved learning advanced concepts like CNN and then taking a deep dive into 2D and 3D models, finally using the nnU-net model proving that it's by far the best model to ever been used for segmentation for this particular dataset.

The journey had multiple stages. I started with 2D image segmentation exploring various models and implementing 2 out of all explored. Then, I transitioned towards advanced dataset involving volumes of images and labels. I achieved considerably competitive evaluation metrics through this model which was able to outperform all the models used with this dataset. The reliability on this model for segmenting complex tumor structures was depicted with the performance it gave.

With such performance, this model can certainly be integrated into the world of medicine. It automates the repetitive task and helps the radiologists diagnose the cancer in its early stages and saves their time for decision making.

All in All, this project came up to be an effective solution in the field of medicine. Not only that, it paved the way towards strong foundational learning experience where I learned much about deep learning concepts and techniques. It also gave me a chance to try to do something impactful for the society.



## 5.1 | Future expansion

There are many doors open for enhancements and exploration:

- **Training on More Diverse Datasets:** There are many other breast datasets available, like DUKE, ISPY-1, ISPY-2, etc. These datasets could be used during model training for making it more generalized.
- **Clinical Validation and Expert Feedback:** The model outputs can be shown to the experts for gaining practical insights in order to refine the output more.
- **Building an interactive software:** A streamlit based web application could be deployed where the experts just need to drop the images there and get the segmentation output.
- **Integration into PACS:** As discussed earlier, this model could be integrated in the PACS system in order to automatically segment the patient's stored MRI images.
- **Inference Optimization:** As the segmentation of the new data would use inference part of the pipeline, optimizing it will make sure that the model can be used in the resource constrained areas.
- **Extension to Classification:** Once segmentation is done, the task of the model could even be extended to classification, where the detected tumors can automatically be classified into benign or malignant case.

# Bibliography

- [1] F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Koehler, T. Norajitra, S. Wirkert, and K. H. Maier-Hein, “nnu-net: Self-adapting framework for u-net-based medical image segmentation,” 2018.
- [2] “BC-MRI-SEG: A breast cancer mri tumor segmentation benchmark,” in *2024 IEEE Conference Publication*, 2024. Accessed: 2025-04-21.
- [3] X. Zhao, Y. Liao, J. Xie, X. He, S. Zhang, G. Wang, J. Fang, H. Lu, and J. Yu, “Breastdm: A dce-mri dataset for breast tumor image segmentation and classification,” *Computers in Biology and Medicine*, vol. 164, p. 107255, 2023.
- [4] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, “nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation,” *Nature Methods*, vol. 18, no. 2, pp. 203–211, 2021.

# Breast Cancer Segmentation using NN-UNet

## ORIGINALITY REPORT

8%

SIMILARITY INDEX

6%

INTERNET SOURCES

4%

PUBLICATIONS

4%

STUDENT PAPERS

## PRIMARY SOURCES

1

Submitted to University of Canberra

Student Paper

1%

2

Submitted to Amsterdam hogeschool voor de kunsten

Student Paper

1%

3

Submitted to Higher Education Commission Pakistan

Student Paper

1%

4

Aashish C. Gupta, Guillaume Cazoulat, Mais Al Taie, Sireesha Yedururi et al. "Fully automated deep learning based auto-contouring of liver segments and spleen on contrast-enhanced CT images", Scientific Reports, 2024

Publication

<1%

5

docslib.org

Internet Source

<1%

6

arxiv.org

Internet Source

<1%

7

Radiful Islam, Rashik Shahriar Akash, Md Awlad Hossen Rony, Md Zahid Hasan. "SAMU-

<1%