Breast Cancer Segmentation using U-Net

Abdul Bhutta
Electrical and Computer Engineering
Toronto Metropolitan University (TMU)
Toronto, Canada
abdul.bhutta@torontomu.ca

Abstract—**Cancer is the second leading cause of death in the world, and breast cancer is one of the most common cancers among it. World Health Organization (WHO) stated that around 2.3 million women were diagnosed with breast cancer while 685,000 died from it. Furthermore, by the end of 2020, 7.8 million women were diagnosed with cancer that are alive within the last five years of 2020, making it the world's most common cancer in the world. To aid the radiologist in detecting the tumour, deep learning models can be implemented to detect and segment the tumour in the breast. This paper will look at the various versions of U-Net models that can be optimized to detect and segment the tumour in an ultrasound image. U-Net is a deep learning model used for image segmentation for biomedical images, and it is one of the best models to implement when the dataset is limited. Furthermore, the current research and U-Net models implemented in similar areas will be explored and analyzed to create an optimized model. The model training will be based on the image size, loss function, and whether to include the normal class. The best model for accuracy and dice similarity coefficient was model 3, which used a custom loss function and image size of 128 by 128 pixels while scoring a dice similarity coefficient of 0.87 with an accuracy of 98%.**

I. Introduction

One of the most dangerous and common cancers among women is breast cancer; while it is not as common in males, they still have a very low chance of getting it. Canada Cancer Society reported in 2023 that it was estimated that 29 400 women and 260 men new cases of breast cancer would be diagnosed. Out of the 29 660 cases, it is predicted that 5400 women and 55 men will die from breast cancer. It is difficult to detect breast cancer in the early stages as it does not show any symptoms, and the tumour may be tiny, which even a radiologist may not be able to detect. There are two types of cancer: Benign and Malignant. Benign cancer is slow-growing and does not spread to other parts of the body. They can be removed through surgery and are a low risk to the patient, generally not life-threatening. However, malignant cancer is extremely dangerous and can spread at a fast rate to other parts of the body even after treatments such as chemotherapy, radiation, and surgery. It traverses through the blood and lymphatic system to different parts of the body while invading nearby tissues and organs, which is the hallmark of cancer and causes damage to the body. Regardless of whether the tumour is benign or malignant, it is crucial to detect the tumour in the early stages to prevent the cancer from spreading to other parts of the body for a higher chance of survival. One of the ways to detect a tumour that can cause breast cancer is through a mammogram or ultrasound. Regardless of the image type, the radiologist must go through many images to detect the tumour, which can be time-consuming and prone to human error. Nonetheless, the radiologist may miss the tumour in the image, which can be life-threatening for the patient. A deep learning model can help the radiologist detect the tumour while reducing the time it takes to detect the tumour and human error. However, in the later section, we will investigate why true positives and false positives are essential in detecting the tumour in the breast. In this paper, we will explore various U-Net models implemented in the research to detect and segment the tumour in the breast while trying to improve the accuracy and efficiency of the model. The model implemented will be a deep learning model, U-Net, an image segmentation model widely and initially designed for biomedical images that allows us to train the model where the dataset is limited. The paper looks to explore the impact of various image sizes, 128x128 and 256x256, while using two different loss functions, binary cross entropy and a custom loss function that combines binary cross entropy and dice loss, to determine the best model for such tasks. One of the primary applications of this model can be to aid radiologists and doctors in detecting the tumour in the early stages or even before any symptoms show.

II. Literature Review

The section explores various papers and implementations conducted using the U-Net model to determine the best model for detecting and segmenting the tumour in the breast. A proposed architecture by Cirean et al. [1] was one of the first deep learning models to achieve significant success in biomedical image processing for neuronal membranes in Electron Microscopy images. A deep neural network classifier was trained to determine whether each pixel in an image was part of a membrane by analyzing the pixel's raw intensity values. The model was implemented using a sliding window technique to train the network on local patches around each pixel. This approach increased the training time required to train a model and the amount of training data but was slow and suffered from redundancy due to overlapping patches. Despite its limitations, the model achieved significant success in the EM segmentation challenge at ISBI 2012, prompting more recent methods that integrate multi-layer features to improve localization accuracy while maintaining contextual awareness.

This led to the development of a deep-learning model, the U-Net architecture, by Ronneberger et al. [2]. It involved fully convolutional networks that were trained end-to-end from very few images and achieved high accuracy in the ISBI cell tracking challenge. The architecture involved a downsampling and upsampling path that could be used on a limited dataset to achieve high accuracy in image segmenting. The network consists of a contracting (downsampling) path and an expansive (upsampling) path. The upsampled output is concatenated with the corresponding cropped feature map from the downsampling path to provide high-resolution features to the decoder. It consists of 3x3 convolutions, a rectified linear unit (ReLU), and a 2x2 max pooling layer for downsampling. The feature channels are doubled after each max pooling layer in the downsampling path. In contrast, the feature channels are halved after each upsampling path and merged with the downsampled features from the contracting path, forming a U-shaped architecture. The U-Net architecture outperformed all the previous state-of-the-art models by a significant margin while setting new records without pre- or post-processing using limited data, as this verifies why the U-Net architecture is one of the best models for biomedical image segmentation.

The U-Net model can perform various tasks such as image segmentation, super-resolution, and denoising. It can perform image segmentation on multiple images, such as brain MRI, breast ultrasound, the nuclei, or even nerve detection images. Anesthesia is one of the most common procedures administered to patients to help with the pain during surgery. A local procedure is used to administer anesthesia rather than total anesthesia. The local procedure is administered through regional anesthesia, which is the injection of local anesthetic near the nerve to limit the pain. However, this procedure comes with a risk of damaging the nerve, block failure, and anesthesia toxicity. A newer technique was established to locate the nerve, called ultrasound-guided regional (UGR), which requires the location of the nerve. Kakade et al. [3] explored using the U-Net model to identify the nerve in the ultrasound image and improve needle tracking. It involved removing noise from the image through the Gaussian filter using the Gaussian smoothing operation, which eliminates the noise by blurring the image after the K-means clustering is applied to determine the background and foreground. However, after applying the Gaussian filter, the image had negligible noise. Another filter was applied, the Gabor filter, a linear filter used to detect the edges in the images and extract the features of the image. The U-Net model was selected as one of the models for the task and training using 5635 images with masks manually annotated by experts for training and another 5508 images for testing. The research yielded promising results, with the U-Net model surpassing all other methods. It achieved a dice similarity coefficient of 0.69, a testament to its superior nerve identification and needle-tracking performance.

Nuclei segmentation is another area of research where segmentation is required. However, brain tumours are hard to detect and segment due to the complexity of the brain and the various shapes and sizes of the tumour. The paper by Vuola et al. l, [4] implemented two deep learning models, Mask R-CNN and U-Net model, which generated a new ensemble model to detect and segment the nuclei in the image. Mask R-CNN is excellent, for instance segmentation, where the model can detect multiple objects of the same class in the image and segment them using a bounding box and mask. At the same time, U-Net is great for semantic segmentation, which involves labelling each pixel in an image with a class. Still, it does not differentiate between different or unique instances of the same class. The ensemble model combines the strengths of both models to improve the model's accuracy in detecting and segmenting the nuclei. It was concluded that the mask R-CNN model outperformed the U-Net model in detecting all the nuclei in the image more accurately than the U-Net model. Still, the U-Net model outperformed the Mask R-CNN model in segmenting the nuclei at a higher precision. However, the ensemble model outperformed both methods with an IoU threshold of 0.7. Combining the strengths of both models is an interesting approach, which leads to the question of whether similar results can be anticipated in different research areas, such as breast ultrasound images.

Segmentation is also essential in research focused on detecting brain tumours. However, brain tumours are hard to detect and segment due to the complexity of the brain and the various shapes and sizes of the tumour. A research paper by Mortazavi-Zade et al. [5] attempted to segment the tumour in the brain using the U-Net model. The model was trained on the BraTS 2018 and 2015 dataset, consisting of 285 and 274 images, respectively. The original U-Net was modified, and the loss function was altered to create a new architecture called the U-Net++. Modifications were made to the skip connections to include nested skip connections, which allowed the model to learn more complex features and patterns in the image for brain segmentation. The loss function was altered to include the dice loss and binary cross entropy loss to improve the accuracy with the backbone of InceptionV3. The results showed the model slightly outperformed the original U-Net architecture with a dice similarity coefficient of 0.90 for the BraTS 2018 dataset and 0.89 for the BraTS 2015 dataset.

A similar study [6] took a unique approach by focusing on validating a novel method for segmentation on the Brats 2015 dataset. This dataset, consisting of 220 high-grade glioma and 54 low-grade glioma images with various sizes and shapes, was manually segmented to create the ground truth mask. The study used a modified U-Net, but its decision not to involve data augmentation set it apart from most traditional approaches. Instead, it leveraged the model's learning ability with fewer data points. The model's performance was evaluated using the dice similarity coefficient, and the results were impressive, with the model achieving the most accurate segmentation with a dice similarity coefficient of 0.90 for a complete tumour, 0.82 for the core tumour and 0.87 for the enhancing tumour.

Breast cancer is the most common cancer in women around the world, and it is the second leading cause of death in the world. Mridha et al. [7] proposed a deep learning model to aid in detecting the tumour in the breast using the U-Net model but with minor modifications to include attention gates where it focuses on specific regions of the image to improve the accuracy of the segmentation and on areas that are important for the segmentation task. They are helpful in cases with a cluttered background or when the object is small in the image. It involved a dataset with 780 images of mammograms and masks with labels normal, benign, and malignant with a size of 500x500 pixels. The dataset was split into training and testing sets, with 90% of the data used for training and 10% used for testing. Before the model's training, preprocessing steps were taken to prepare the dataset for the model training. The normal class was dropped from the dataset as it does not contain any information about the tumour, and the model will not learn anything from it. Furthermore, the images that contained multiple masks were combined into a single mask by taking the maximum pixel value at each pixel, which ensured the mask represented the whole tumour. The model was evaluated based on the accuracy, precision, recall, and f1 score, where the overall accuracy of the model was 95%.

Mammograms are not the only way of detecting the tumour in the breast; they can also be detected through magnetic resonance imaging (MRI). Although MRI is not used for routine screening but rather for high-risk patients or patients with dense breast tissue. It is time-consuming and requires radiologists to segment each image to detect the tumour manually. Neural networks and deep learning models can be great tools for helping the radiologist and sometimes may even surpass the radiologist in detecting the tumour. The paper by Yue et al. [8] proposed a deep learning model with a fully convolution neural network called Res U-Net that adds a residual block to the encoder portion and skip connections of the original U-Net model to detect tumour through MRI images. It is used to extract features and may help solve the vanishing gradient problem in the network caused by deep-layer networks. The dataset, which contained 1000 images of various sizes, was split into 80% training and 20% testing while using the cross entropy as the loss function and the Adam optimizer to train the model. The model outperformed all other models for the segmentation task and achieved a dice similarity coefficient of 0.89.

The research conducted by Matic et al. [9] involved using various pre-trained models such as VGG16, ResNet, and InceptionV3 to extract the features of the image while implying the transferred learning should outperform the model trained from scratch as the pre-trained models have been trained on a larger dataset that will help the model learn the features of the image. Before training the model, A few data augmentations were applied to the dataset to improve the model's generalization. It included rotation, flipping, and scaling of the images to create new images to be appended to the current dataset. However, the most significant impact was modifying the encoder path, which downsamples the image to extract the features and applies a transfer learning technique. It also stated that only the classes with the tumour should be included in the model training, while the normal class should be dropped as it does not contain any information about the tumour.

III. Problem Statement

The common occurrence of breast cancer in the world is alarming, and we need to find a way to automate the process of detecting and segmenting the tumour in the breast to help radiologists and doctors in the early stages. One of the most challenging tasks in detecting tumours is manually segmenting the tumour from the background in an ultrasound image with the naked eye. It is difficult as the tumour can be of any size, shape, and texture. Furthermore, the radiologist may miss the tumour in the image, and in the long run, it can be life-threatening to the patient as cancer may spread to other parts of the body to a point where it is untreatable or may cause death. A deep learning model can help aid the radiologist in an automated way to detect the tumour while reducing the time it takes to detect the tumour and reducing human error. An automated application can accept multiple images sent through the model, which detects the tumour in the image and segments the tumour from the background. Whether the application is used during the day or night, the segmented images will be stored or sent to the radiologist for further inspection and diagnosis. However, it must be stated that a deep learning model is not a replacement for a radiologist but rather an aiding tool to inspect further the images that the radiologist may have missed or to confirm the diagnosis while reducing the time it takes to detect the tumour.

IV. System Model

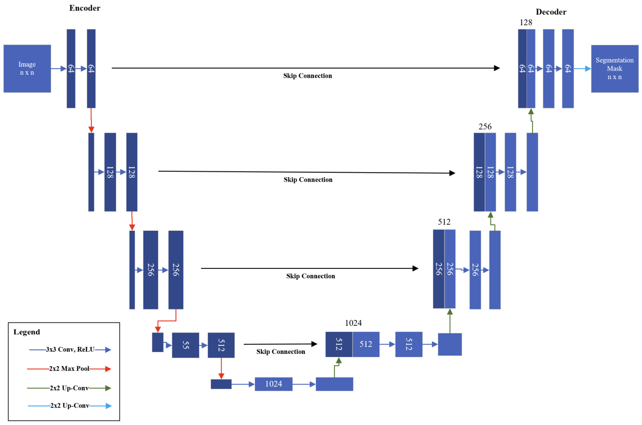

Fig. 1. U-Net Architecture

The U-Net architecture, a groundbreaking innovation, was initially crafted to create models for biomedical images with limited data. It excels in generalizing the model to unseen data while maintaining high accuracy and speed. The model's purpose is to detect and segment the tumour in the breast, regardless of the tumour size or the type of cancer, transforming it into a binary classification problem. The model accepts a vector form of an image and produces a segmentation mask, which is compared to the ground truth mask to compute the loss. This loss is then used to adjust the network parameters by minimizing the loss function through backpropagation and enhancing the model's accuracy by updating the network weights.

The U-Net architecture, with its symmetrical structure, is a powerful tool in biomedical image segmentation. It consists of two parts: the encoder and the decoder. The encoder, a downsampling path, extracts features or context from the image. Simultaneously, the decoder, an upsampling path, localizes the object in the image and upsamples the intermediate features to the original image resolution, producing the final output with the segmentation mask. The encoder and decoder paths are connected by skip connections, creating a U-shape, which gives the architecture its name, U-net. This symmetrical design and architecture is illustrated in Figure 1.

The encoder and decoder consist of convolutional, max-pooling, and upsampling layers. It is a Convolutional Neural Network (CNN) using an encoder-decoder architecture. It extracts valuable features from an image, such as recognizing objects like tumours, cars, or people. The features extracted are passed through the encoder consisting of repeated blocks of convolutional layers and max pooling layers to extract intermediate features. In the early layers of the model, the low to mid-level features, such as the edges, textures, contours, and shapes, are extracted, while in the deeper layers, the high-level features, such as the context and semantic information, are extracted. Once an object is identified using the encoder path, the decoder uses these features to get back to the original image resolution to localize the object at the pixel level where the object resides in the original image. The extracted features are then upsampled by a corresponding decoder, where saved copies of the encoder features are concatenated or added to the decoder features through skip connections.

There are two types of connections between the encoder and decoder path: bottleneck and skip connections. The connecting path skip connection takes a copy of the features extracted by the encoder and concatenates them to the symmetrical part at the same decoder stage. This means the features are placed alongside the decoder features simultaneously, meaning the convolutional layers can operate on both the encoder and decoder features. This is because the decoded features might include more semantic information, such as an area containing a car or a person. In contrast, the encoded features contain more spatial information, such as the pixels of where the object resides, to detect the objects' edges, textures, and shapes. Combining spatial and semantic information from the encoder and decoder paths allows the network to refine the segmentation to produce a pixel-perfect segmentation mask at the output.

The final layer of the decoder is a 1x1 convolutional layer, a point-wise convolution, that passes through a sigmoid activation function to produce a probability to classify each pixel into two classes: background (0) or object (1). The output is a segmentation mask where each pixel is either a 0 or 1 representation of one of the classes. The loss function generally used for binary classification problems is the binary cross-entropy loss, computed to compare the predicted mask to the ground truth mask while applying the backpropagation algorithm to adjust the network parameters to minimize the loss and improve the model's accuracy. The exact architecture implemented and evaluated in this paper is discussed further in the methodology section, where it discusses the layers, input shape, output shape, channels, and filter size of a 128x128 image size model.

V. Methodology

A. Dataset

The dataset used for this research is the Breast Ultrasound Image Dataset from Kaggle, which consists of 780 images of size 500x500. The dataset is from 2018 and collected from 600 female patients aged 25 to 75. It consists of three classes: Normal, Benign, and Malignant. Whether the tumour detected is benign or malignant, the tumour is classified as a cancerous tumour. Due to this, the comparison of including the normal class and not including it will be explored to see its impact on the accuracy of the model detecting a tumour. If the normal class is included in the model's training process, it may introduce noise and irrelevant information while training the model. There are only 266 images of the normal class, which needs to be improved to train the model effectively. The dataset is imbalanced, and the normal class is underrepresented, which may affect the model's accuracy in detecting the tumour in the image. The U-net model will be trained using both methods to determine the best model for detecting and segmenting the tumour in the breast to verify the assumption.

B. Preprocessing

A few preprocessing steps were taken to prepare the dataset for the model training. Initially, the entire dataset was resized to 128x128 or 256x256 to determine the impact of the image size on the model's accuracy in detecting a tumour. Furthermore, the images were converted to grayscale to reduce the computational cost of training the model and to focus on the relevant features of the tumour in the image. Lastly, the images were normalized to a range of 0 to 1 to improve the convergence of the model during training. The images containing multiple masks for the same image were combined into a single mask to improve the model's accuracy in detecting various tumours in the image. The masks were combined by adding the pixel values of the masks together and then thresholding the pixel values to 0 or 1 to create a binary mask. All the images and masks were stored in a NumPy array to train the model. Once the images were preprocessed, the dataset was split into training, validation, and test sets. The main difference here is each model was trained on the same dataset even after shuffling, as the shuffling included a parameter called `random state` to ensure the dataset was shuffled in the same way for each model to ensure reproducibility. This was done to ensure each model was trained on the same dataset to compare the results accurately to verify the impact of the image size and loss function on the model's accuracy in detecting the tumour. The training set consisted of 90% of the dataset, while the validation and test set consisted of 5% of the dataset each. The training set was used to train the model, the validation set was used to evaluate during training to see how well the model is generalizing to unseen data, and the test set was used to assess the model's accuracy after training to predict the mask segmentation.

C. Model Architecture

The current model implemented consists of a total of 29 layers that are divided into the encoder and decoder path with skip connections and bottleneck connections. The encoder path consists of multiple convolution layers, which are 3x3 filters with a stride of 1, and a ReLu activation function is applied to each of the features in the convolutional layer. The max pooling layer consists of a 2x2 filter with a stride of 2 and is used at each stage of the encoder to downsample the features to reduce the spatial dimensions of the image. To compensate for the loss of spatial information, the feature channels are doubled after each max pooling layer to increase the context information at each encoder stage. The channels allow the network to learn more complex features and patterns in the image and help determine various shapes and textures. The decoder consists of 3x3 convolutional layers, each followed by a ReLu activation function. Instead of max pooling layers downsampling the features, the decoder upsamples the current set of features by applying a 2x2 convolutional layer, and then the number of channels is halved. This comprehensive approach allows the model to recover the spatial information lost during the downsampling, ensuring a secure and complete understanding of the image's features. The implementation of the U-Net architecture and the layers is shown in Table 1.

TABLE I
U-NET LAYERS

Layer	Name	Input Shape	Output Shape	Filter Size
0	input	(128, 128, 1)	(128, 128, 1)	1x1
1	conv2d_1	(128, 128, 1)	(128, 128, 64)	3x3
2	conv2d_2	(128, 128, 64)	(128, 128, 64)	3x3
3	max_pooling2d_1	(128, 128, 64)	(64, 64, 64)	2x2
4	conv2d_3	(64, 64, 64)	(64, 64, 128)	3x3
5	conv2d_4	(64, 64, 128)	(64, 64, 128)	3x3
6	max_pooling2d_2	(64, 64, 128)	(32, 32, 128)	2x2
7	conv2d_5	(32, 32, 128)	(32, 32, 256)	3x3
8	conv2d_6	(32, 32, 256)	(32, 32, 256)	3x3
9	max_pooling2d_3	(32, 32, 256)	(16, 16, 256)	2x2
10	conv2d_7	(16, 16, 256)	(16, 16, 512)	3x3
11	conv2d_8	(16, 16, 512)	(16, 16, 512)	3x3
12	max_pooling2d_4	(16, 16, 512)	(8, 8, 512)	2x2
13	conv2d_9	(8, 8, 512)	(8, 8, 1024)	3x3
14	conv2d_10	(8, 8, 1024)	(8, 8, 1024)	3x3
15	conv2d_11	(8, 8, 1024)	(8, 8, 512)	3x3
16	up_sampling2d_1	(8, 8, 512)	(16, 16, 512)	2x2
17	concatenate_1	(16, 16, 512)	(16, 16, 1024)	3x3
18	conv2d_12	(16, 16, 1024)	(16, 16, 512)	3x3
19	conv2d_13	(16, 16, 512)	(16, 16, 512)	3x3
20	up_sampling2d_2	(16, 16, 512)	(32, 32, 512)	2x2
21	concatenate_2	(32, 32, 512)	(32, 32, 1024)	3x3
22	conv2d_14	(32, 32, 1024)	(32, 32, 256)	3x3
23	conv2d_15	(32, 32, 256)	(32, 32, 256)	3x3
24	up_sampling2d_3	(32, 32, 256)	(64, 64, 256)	2x2
25	concatenate_3	(64, 64, 256)	(64, 64, 512)	3x3
26	conv2d_16	(64, 64, 512)	(64, 64, 128)	3x3
27	conv2d_17	(64, 64, 128)	(64, 64, 128)	3x3
28	up_sampling2d_4	(64, 64, 128)	(128, 128, 128)	2x2
29	conv2d_18	(128, 128, 128)	(128, 128, 64)	3x3
30	conv2d_19	(128, 128, 64)	(128, 128, 1)	1x1

D. Loss Function

The models have been trained using two distinct loss functions. The first, binary cross-entropy, is a widely used loss function in biomedical image segmentation. This is because the segmentation task is essentially a binary classification problem, where each pixel is classified as either background or object. In contrast, each pixel is associated with a label of 0 or 1 after the sigmoid activation function is applied to the output layer, which gives a probability of each pixel. It is used to compute the loss between the predicted and ground truth masks to adjust the network parameters by minimizing the loss function to improve the model's accuracy. The binary cross entropy loss is computed using the following formula:

$$L_{BCE}(p, p_i) = -\frac{1}{N}\sum_{i=1}^{N} p_i \log(p_i) + (1 - p_i)\log(1 - p_i) \quad (1)$$

where y is the ground truth mask, $\hat{y}$ is the predicted mask, $|y \cap \hat{y}|$ is the intersection of the ground truth mask and the predicted mask,

The second loss function is a custom loss function that combines the binary cross entropy loss and the dice loss. The dice loss is a loss function used for image segmentation problems and is computed by taking the intersection over the union of the predicted mask and the ground truth mask. This means the predicted and ground truth masks are compared to calculate the loss, and the loss is minimized while the network parameters are updated. The dice loss is calculated using the following formula:

$$L_{DiceLoss}(y, \hat{y}) = 1 - \frac{2 \times |y \cap \hat{y}|}{|y| + |\hat{y}|} \quad (2)$$

where y is the ground truth mask, $\hat{y}$ is the predicted mask, $|y \cap \hat{y}|$ is the intersection of the ground truth mask and the predicted mask,

The dice loss optimizes the dice similarity coefficient (DSC), a metric that calculates the similarity or the area of interest between the predicted and ground truth. The dice loss is excellent for imbalanced datasets where the number of pixels in the background is much larger than the number of pixels in the object or when there is a need for a high overlap accuracy between the predicted mask and the ground truth mask, such as medical imaging. Combining the dice and binary cross entropy loss can help train the model by addressing pixel-wise and region overlap accuracy. The loss function is minimized when the predicted and ground truth masks are similar or penalized when dissimilar, even though the accuracy at the pixel level is high. The two losses are added together to counter the limitations of each loss function and improve the accuracy of the model. The custom loss function is computed using the following formula:

$$L_{CL}(y,\hat{y}) = L_{BCE}(y,\hat{y}) + L_{DiceLoss}(y,\hat{y}) \tag{3}$$

where $L_{CL}(y,\hat{y})$ is the custom loss function, $L_{BCE}(y,\hat{y})$ is the binary cross entropy loss, and $L_{DSC}(y,\hat{y})$ is the dice loss.

E. Training

All models were trained on Google Colab using Colab Pro service that allowed access to Nvidia A100 GPU with 40GB memory. This was a costly service, but training the model on a premium GPU was necessary to reduce the model's training time. The compute units for an A100 GPU are around 11.4 compute/hour, and the Google Pro service cost is around 15 dollars for 100 compute hours. To this day, over 300 compute units have been depleted for this research purpose only. The model was trained using the Adam optimizer with a learning rate of 0.001 and a default batch size of 32. Increasing the batch size value to 64 and greater was crashing the kernel as the GPU ran out of memory. During training, two custom callbacks were used to monitor the training process. The first callback was the ModelCheckpoint callback shown below,

```
model_checkpoint = tf.keras.callbacks.
    ModelCheckpoint('
    model1_unet_binaryloss_128.h5',
    save_best_only=True)
```

that saved the model with the lowest validation loss as the best model and saved it as an h5 file.

The second callback used the TensorBoard callback, as shown below, to process and visualize the training and validation loss during training. This allowed us to see the training and validation loss at each epoch to verify if the model was generalizing to unseen data.

```
tensorboard_callback = tf.keras.callbacks.
    TensorBoard(log_dir = 'logs',
    histogram_freq = 1)
```

Each model was trained for 100 epochs with the two callbacks integrated into the model.fit() function to save the best model and monitor the training process as shown below,

```
model_history = model.fit(X_train, y_train,
    epochs = 100, validation_data = (X_val,
    y_val), callbacks=[tensorboard_callback,
     model_checkpoint])
```

The best models for the two image sizes, loss functions, and the normal class included and not included were saved on Google Drive to prevent the model from being lost if the runtime was disconnected and was used for inference.

VI. Results

The training process for each model varied based on the image size, as the model with a 128x128 image size trained faster than the model with a 256x256 image size. The 128x128 images took approximately an hour to train, and the 256x256 took around 80 minutes for 100 epochs on the A100 GPU. Each epoch training was visible through the TensorBoard callback, which showed the training and validation loss. The best model based on accuracy and the dice similarity coefficient was model 3, which had a dice similarity coefficient of 0.874. Each image shown in the result section is from model 3, but further inspection can be done for each model through the Jupyter Notebook on GitHub. The model with the normal class integrated into the dataset (model 5) outperformed all the models except for model 3 regarding the dice similarity coefficient and accuracy. Models three and five had a few false positives, which classified the background as a cancerous tumour in the image, which was not cancerous. This is a common problem in medical imaging where the model may classify the background as a cancerous tumour. Still, having a few false positives rather than false negatives is better. If the model does not pick up the tumour and classifies it as the background, it can be life-threatening to the patient. The research also highlighted the superiority of the custom loss function over the binary cross entropy loss function. The custom loss function consistently detected tumours at a higher dice similarity coefficient, indicating its effectiveness in this context. While the image size had a minor impact on the model's accuracy, it showed a trend where the 128x128 image size performed slightly better than the 256x256 image size. This could be attributed to the model learning more features in the image at a lower resolution, especially when the dataset is limited. Additionally, the field of view assisted the model in focusing on relevant information in the image. These findings suggest that the custom loss function and the 128x128 image size are optimal for detecting and segmenting tumours in the breast.

A. Test Results

The output from the model for the first test image is shown in Figure 2. The image on the left is the original image, the image in the middle is the ground truth mask, and the image on the right is the predicted mask.

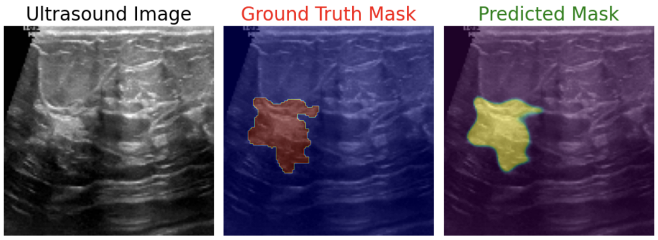

Fig. 2. Inference with Predicted Output

Table 2 provides the properties of each model, including image size, loss function, and if the normal class was included in the dataset.

TABLE II
MODEL PROPERTIES

Model	Image size	Loss Function	Normal Data
Model 1	128x128	Binary Cross Entropy	No
Model 2	256x256	Binary Cross Entropy	No
Model 3	128x128	Custom Loss Function	No
Model 4	256x256	Custom Loss Function	No
Model 5	128x128	Custom Loss Function	Yes

In Table 3, you will find all the essential information about the test set results from all the models. This comprehensive overview includes the model, accuracy, precision, recall, and dice similarity coefficient, which are crucial for understanding and comparing the models' performance.

TABLE III
MODEL EVALUATION

Model	Dice Similarity Coefficient	Accuracy	Precision	Recall
Model 1	0.666	0.956	0.711	0.634
Model 2	0.666	0.957	0.696	0.640
Model 3	**0.874**	**0.984**	**0.896**	**0.880**
Model 4	0.653	0.955	0.671	0.640
Model 5	0.779	0.967	0.785	0.773

*Model 3 had the best dice similarity coefficient of 0.874

The confusion matrix in Figure 3 shows the true positive, false positive, true negative, and false negative of the best model, model 2, for each pixel in the image. The true positive is the number of pixels that are correctly classified as the background, the false positive is the number of pixels that are incorrectly classified as a tumour, the true negative is the number of pixels that are correctly classified as the background, and the false negative is the number of pixels that are incorrectly classified as the background.

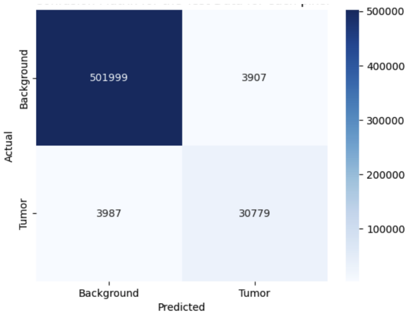

Fig. 3. Confusion Matrix

The dice coefficient is one of the best metrics to evaluate the model's accuracy in detecting and segmenting the tumour in the image. The dice similarity coefficient of the three models based on three categories was computed as shown in Table 4. Dice coefficient below 0.5 is considered poor, between 0.5 to 0.7 is considered moderate, between 0.7 to 0.9 is considered good, and above 0.9 is considered excellent.

TABLE IV
MODEL DICE SIMILARITY COEFFICIENT COMPARISON

Model	Moderate	Good	Excellent
Model 1	25	20	11
Model 2	24	19	10
Model 3	31	30	14
Model 4	22	10	11
Model 5	27	23	11

B. U-Net Model Layers

A 128x128 image with its actual segmentation mask, Figure 4, is passed through the neural network to analyze how each layer of the models performs to get the output as a segmentation mask.

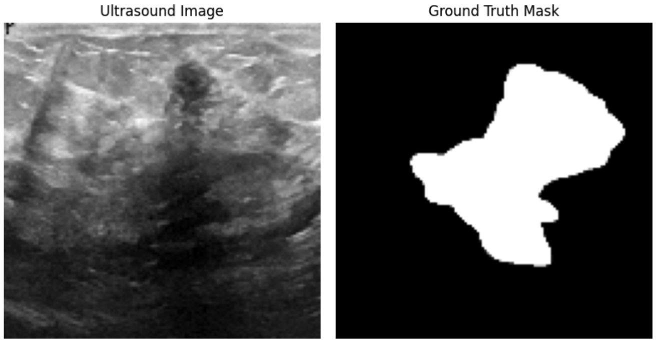

Fig. 4. 128x128 Ultrasound Image

The first layer of the U-Net Architecture is the input layer, which is the image of size 128 by 128 by 1, where the size is 128 by 128, and the number of channels is one because it is a grayscale image. On top of the figure, the image size at the output of each layer is shown in the format `(channel, size, size, filter)` as shown in the figures below. The first 11 layers are part of the encoder path that downsamples the image to extract features from the image, as shown in Figures 5, 6, and 7. The encoder path consists of convolutional layers with 64, 128, 256, and 512 filters with a kernel size of 3x3 and a ReLU activation function. The max-pooling layers with a pool size of 2x2 are applied to each stage of the encoder to reduce the spatial dimensions of the image. The max pooling layers reduce the size of the image by a factor of 2, which is why the size of the image is reduced from 128 by 128 to 64 by 64, 32 by 32, 16 by 16, and 8 by 8. The number of channels is doubled after each max pooling layer to increase the context information at each encoder stage.

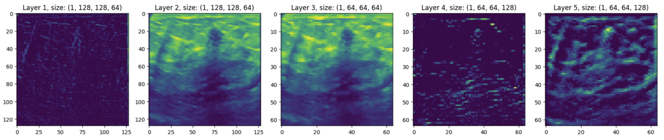
Layer 1, size: (1, 128, 128, 64)
Layer 2, size: (1, 128, 128, 64)
Layer 3, size: (1, 64, 64, 64)
Layer 4, size: (1, 64, 64, 128)
Layer 5, size: (1, 64, 64, 128)

Fig. 5. Layer 1 to 5

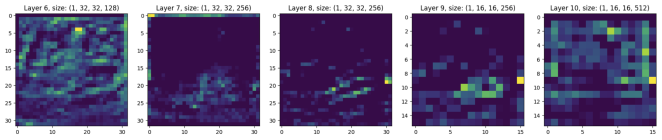
Layer 6, size: (1, 32, 32, 128)
Layer 7, size: (1, 32, 32, 256)
Layer 8, size: (1, 32, 32, 256)
Layer 9, size: (1, 16, 16, 256)
Layer 10, size: (1, 16, 16, 512)

Fig. 6. Layer 6 to 10

Layers 12 to 13 are the bottleneck connections at the bottom of the U-Net architecture, where the encoder switches to the decoder path. The output of the bottleneck connections is shown in Figure 7. The bottleneck layer takes the output of the last max pooling layer and passes it through a 3x3 convolutional layer with 1024 filters and a ReLU activation function. The output of the bottleneck layer is then passed through another 3x3 convolutional layer with 1024 filters and a ReLU activation function to produce the output of the bottleneck layer.

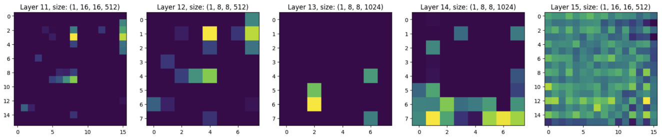
Layer 11, size: (1, 16, 16, 512)
Layer 12, size: (1, 8, 8, 512)
Layer 13, size: (1, 8, 8, 1024)
Layer 14, size: (1, 8, 8, 1024)
Layer 15, size: (1, 16, 16, 512)

Fig. 7. Layer 11 to 15

Layers 14 to 29 are part of the decoder path, which localizes the object in the image and upsamples the intermediate features to the original image resolution, producing the final output with the segmentation mask. As we increase the number of layers in the decoder path, the size of the image increases as we upsample the features to the original image resolution. In contrast, the extracted features are added to the decoder features, as shown in Figures 7, 8, 9 and 10.

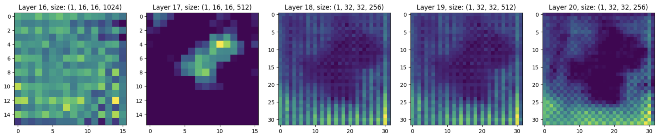
Layer 16, size: (1, 16, 16, 1024)
Layer 17, size: (1, 16, 16, 512)
Layer 18, size: (1, 32, 32, 256)
Layer 19, size: (1, 32, 32, 512)
Layer 20, size: (1, 32, 32, 256)

Fig. 8. Layer 16 to 20

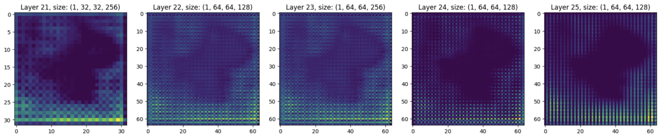
Layer 21, size: (1, 32, 32, 256)
Layer 22, size: (1, 64, 64, 128)
Layer 23, size: (1, 64, 64, 256)
Layer 24, size: (1, 64, 64, 128)
Layer 25, size: (1, 64, 64, 128)

Fig. 9. Layer 21 to 25

The last layer of the decoder is a 1x1 convolutional layer with a sigmoid activation function that produces a probability to classify each pixel into two classes: background (0) or object (1) to display the segmentation mask, as shown in the last image in Figure 10.

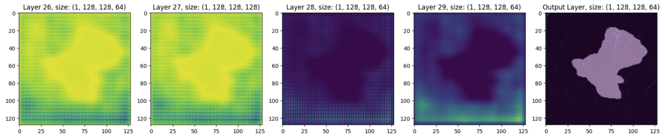
Layer 26, size: (1, 128, 128, 64)
Layer 27, size: (1, 128, 128, 128)
Layer 28, size: (1, 128, 128, 64)
Layer 29, size: (1, 128, 128, 64)
Output Layer, size: (1, 128, 128, 64)

Fig. 10. Layer 26 to 29 and Output Layer

VII. Conclusion

The U-Net model was trained on the Breast Ultrasound Image Dataset to detect and segment the tumour in the breast. The dataset was preprocessed to prepare the images for training, and the model was trained using two loss functions: binary cross entropy and a custom loss. The model was evaluated based on the dice similarity coefficient and accuracy. The results showed that the custom loss function outperformed the binary cross entropy loss function, and the model trained on 128x128 images performed better than the model trained on 256x256 images. The model that included the normal class in the dataset outperformed most except the best model. The best model achieved a dice similarity coefficient of 0.874, an accuracy of 0.984, a precision of 0.896, and a recall of 0.880. Future work could involve training the model on a larger dataset to improve the model's accuracy and generalization to unseen data. Additionally, the model could be fine-tuned using transfer learning or data augmentation techniques, which can be applied to improve the model's performance on the Breast Ultrasound Image Dataset.

References

[1] D. Cirean, A. Giusti, L. M. Gambardella, and Schmidhuber, “Deep neural networks segment neuronal membranes in electron microscopy images,” *Proceedings of Neural Information Processing Systems*, vol. 25, 01 2012.

[2] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[3] A. Kakade and J. Dumbali, “Identification of nerve in ultrasound images using u-net architecture,” in *2018 International Conference on Communication information and Computing Technology (ICCICT)*, 2018, pp. 1–6.

[4] A. O. Vuola, S. U. Akram, and J. Kannala, “Mask-rcnn and u-net ensembled for nuclei segmentation,” in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, 2019, pp. 208–212.

[5] S. A. Mortazavi-Zadeh, A. Amini, and H. Soltanian-Zadeh, “Brain tumor segmentation using u-net and u-net++ networks,” in *2022 30th International Conference on Electrical Engineering (ICEE)*, 2022, pp. 841–845.

[6] T. Yang and J. Song, “An automatic brain tumor image segmentation method based on the u-net,” in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, 2018, pp. 1600–1604.

[7] K. Mridha, T. Sarker, S. D. Bappon, and S. M. Sabuj, “Attention u-net: A deep learning approach for breast cancer segmentation,” in *2023 International Conference on Quantum Technologies, Communications, Computing, Hardware and Embedded Systems Security (iQ-CCHESS)*, 2023, pp. 1–6.

[8] W. Yue, H. Zhang, J. Zhou, G. Li, Z. Tang, Z. Sun, J. Cai, N. Tian, S. Gao, J. Dong, Y. Liu, X. Bai, and F. Sheng, “Deep learning-based automatic segmentation for size and volumetric measurement of breast cancer on magnetic resonance imaging,” *Frontiers in Oncology*, vol. 12, p. 984626, 08 2022.

[9] Z. Matic and S. Kadry, “Tumor segmentation in breast mri using deep learning,” in *2022 Fifth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU)*, 2022, pp. 49–51.