Detection of Region of Interests (RoIs) of Breast Cancer Biopsy using Deep Learning

Nguyen Van Quoc Chuong Jenlarp Jenlarpwattanakul
Wilson Cheng-Han Wang Clancy Heyworth

June 3, 2024

Abstract

Breast cancer diagnosis relies heavily on Hematoxylin and Eosin (H&E) stained image analysis, subject to notable observer variability. This study employs deep learning to mitigate these inconsistencies, using Unet and Unet enhanced with a ResNet50 encoder to segment and classify regions of interest (RoIs) within the BRACS dataset [1]. Our methodology aims to increase diagnostic precision and efficiency in breast cancer pathology, potentially improving patient outcomes by integrating robust deep-learning tools. The study found U-Net inadequate for detecting UDH regions in breast tissue images, although modifications with a ResNet50 encoder showed some promise despite issues with precision. Explorations with a DenseNet model resulted in high classification accuracy which did not transfer when used as an encoder for U-Net.

1 Related Works

- Brancati et al. (2022) [1] describe the construction of the BRACS dataset for carcinoma subtyping in H&E Histology Images. Within their paper, they describe the process for annotation consensus and the other collection mechanisms. This dataset has formed the basis for area of research in various image-based deep learning reports, including our own.
- Ronneberger et al. (2015) [2] detail the UNet segmentation model upon which this report is predicated. The exact architecture of UNet and its beneficial in medical image segmentation is discussed later in the report.
- Hamida et al. (2021) [3] explores the usage of deep learning and UNet in histopathological images analysis of colon cancers. The paper shares similar intentions with this paper, in that it seeks to use UNet to segment histopathological images, albeit for a different form of tissue.

2 Problem & Significance

Breast cancer is among the most frequently diagnosed cancer in women, and the disease can be attributed the greatest number of deaths in women [1]. Hematoxylin and eosin scans (H&E) of patient's breast tissues are a vital tool for pathologists to diagnose cancer [4]. Through the resulting images, pathologists can identify key features that indicate the presence of cancerous cells. However, these scans are often complex and contain ambiguous information that renders it difficult for pathologists to analyse. This process is time-consuming, and there exists a large amount of inter-observer and intra-observer disagreements [1], [5].

To solve this issue, this report seeks to detail the developmental process and strategy towards creating a model that can accurately identify regions of interest within H&E images. Deep learning has the potential to provide an unbiased and efficient way to aid the diagnosis of breast cancer based on H&E images. Accurate and early detection of breast cancer can improve the diagnosis of the disease and lead to a better survival rate [6]. The project aims to utilize deep learning on high quality H&E images, BRACS dataset, to classify Regions of Interest (ROIs) that could aids the diagnosis of breast cancer. ROIs is of particular interest because pathologists begin their assessments at a broader level and then focusing on ROIs [7]. Automatic ROIs classifications using deep learning has the potential to reduce the lengthy process of examine the H&E images.

3 Dataset

The BRACS (BReAst Carcinoma Subtyping) dataset acquired from Brancat at el [1] comprises 547 whole-slide images (WSIs) and 4539 regions of interest (ROIs) from 189 patients, annotated by three expert pathologists. The annotations cover seven lesion types, including Pathological Benign, Usual Ductal Hyperplasia, Flat Epithelial Atypia, Atypical Ductal Hyperplasia, Ductal Carcinoma in Situ, Invasive Carcinoma, and Normal tissue. WSIs are available in .svs format and can exceed 100,000 by 100,000 pixels in resolution, with corresponding .qpdata files for annotations. ROIs are provided as high-resolution .png files, named according to their WSI and lesion type, often surpassing 4,000 by 4,000 pixels .

4 Model choices

4.1 U-net

The task of identifying ROIs in H&E images is a critical step in the diagnosis and study of diseases such as invasive breast cancer. As such Patil at el [7] suggested a technique that utilizes convolutional neural networks (CNNs) to determine when it's necessary to zoom in for a closer examination. A notable advancement in this area is the development of the U-Net model which revolutionized biomedical image segmentation through its efficient use of deep learning for precise delineation of complex structures in medical images [2]. This model has been particularly effective in processing Whole Slide Images (WSIs), demonstrating its robustness and accuracy in extracting detailed features necessary for clinical analysis.

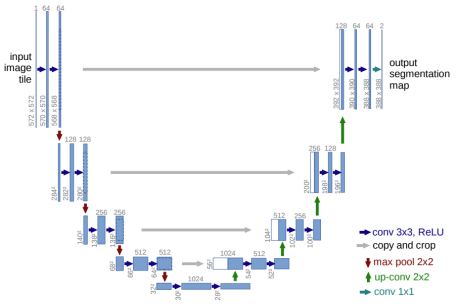


Figure 1: Architecture of the U-Net model used for medical image segmentation [2].

U-Net's design allows the model to segment images based on features both detailed and general [8]. While conventional CNN models excel at extracting image features, U-Net's encoder-decoder structure better allows for preservation of spatial information and is better suited to segment different regions in medical images [8]. This was demonstrated by Baccouch et al [9], where a basic U-Net model and an adapted model achieved 93% and 98% accuracy respectively, compared to 91% accuracy by a CNN model on cardiac MRI scans.

The structure of U-Net is composed of two paths (as seen in figure 1); the first of which is the contracting path, or encoder path, which is arranged into a typical convolution network. The encoder provides classification information on the input. In Base U-Net, each block in this path is that of two successive 3x3 convolution layers followed by a ReLU activation unit and a max-pooling layer. Such blocks can be repeated several times as necessary. The decoder path utilizes the features of the contracting path in up-convolutions and concatenations. Hence, the second path learns localized classification of parts of the images. Each stage of this path upsamples the feature map with a 2x2 up-convolution, then, the feature map from the parallel later in the contracting path is cropped and concatenated onto the new feature map. This skip connection reintroduces detailed features into the decoder, where otherwise U-Net would be limited to extracting more general features. These blocks repeat until a 1x1 convolution layer to produce the segmented image [8].

Our research is inspired by the application of U-Net to segment ROIs in WSIs of invasive breast cancer, as demonstrated by Patil et al [6]. Given the visual and structural similarities between the WSIs [6] and the images in the BRACS dataset compiled by Brancati et al [1], we believe that the U-Net model is exceptionally well-suited for our project. This approach not only leverages the proven

capabilities of U-Net in handling high-resolution H&E images but also aligns with our objective to enhance the accuracy and efficiency of cancerous tissue detection within the BRACS dataset. We aimed to identify the UDH (Usual Ductal Hyperplasia) regions which has the potential to grow into Atypical Ductal Hyperplasia at the ductal region of the breast tissue Brancat at el [1].

4.2 UNet with ResNet Encoder

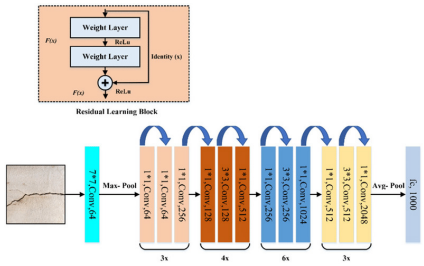


Figure 2: Architecture of a ResNet50 model. [10]

Pretrained encoders for visual tasks are often utilised to leverage their weightings based on prior datasets onto a new task. Through encapsulating such encoders within a larger model, the larger model can theoretically be provided a better basis to learn a specific allocation than from an encoder without pre-trained weights in a process called transfer learning. ResNet50 is a well known convolutional framework, and has been used successfully as an encoder by researchers to construct segmentation models [11]. Weights from ImageNet were used to give the model a generic basis through which it could learn. We will refer this model as SMP through out this report.

4.3 UNet with DenseNet Encoder

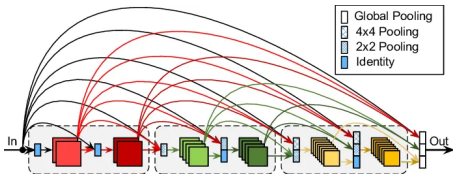


Figure 3: Architecture of a DenseNet121 model. [12]

DenseNet is a convolutional neural network which utilizes forward connections between each block of layers such that every layer is connected outside of the linear forward motion [11].

A DenseNet121 model was trained to classify the tiles as either containing or not containing a UDH region. The weights of this model were then incorporated into a UNet with a DenseNet as the encoder. Unlike the ImageNet-based model of the UNet-ResNet50 encoder which had weights based on generic images, it was proposed that this DenseNet could improve performance through becoming an encoder with specific familiarity in the dataset and would thereby create a basis for the decoder's ability to identify the target regions. Moreover, the possible complexity of the patterns required for a model to recognise could be encapsulated better by DenseNet's ability to maintain contextual information through its skip connections. Similarly to the UNet-Resnet50 combination, researchers have seen success in employing DenseNet in segmentation models [13].

5 Preprocessing and Augmentation

5.1 Generating Tiles

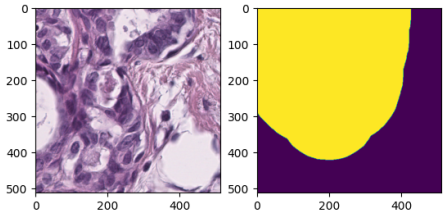


Figure 4: Example Tile and Corresponding Mask

From a practical perspective, training a model on entire slides would be unfeasible due to their immense size. Instead, the slides were split into tiles with dimensions 512 by 512 pixels. The reasoning for these dimensions is that these were the largest size computationally possible for the devices through which we ran our UNet, however it was believed that this would be large enough to retain necessary detail for the model to operate well.

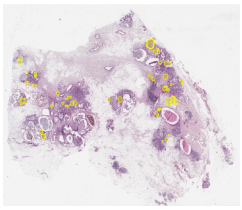
A UNet attempts to predict a binary mask of classes, which necessitates an expected mask output. The ROIs labelled by pathologists were stored within QPData file formats, which were exported to GeoJSON files via the Qupath software, From there, the Pandas library was utilised to intersect the dataframe of annotations with a dataframe of polygons representing the 512 by 512 tiles, and the resulting polygons were rasterized into binary masks for each of our tiles.

5.2 Augmentation of the Dataset

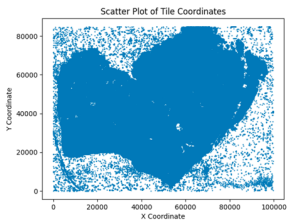
Upon examination of the tiles within the BRACS dataset, it quickly became apparent that regions of interests were sparse within the slides as shown in Figure 5a . This scarcity of tiles featuring regions of interest meant that to train a model on the entire dataset risked rewarding classifying everything as without a region of interest. The necessary workaround then was to artificially reduce the number of tiles included which did not feature a region of interest.

To increase the number of tiles within this new curated dataset, augmentations were applied to the image matrices such that they were randomly flipped horizontally and vertically, along with the accompanying mask. The benefits of augmentation on small datasets for image classification models is well established [14], and it was hoped that in doing so the possible issue of few regions of interests could be avoided. The image matrices from tiles were flipped horizontally then vertically with a probability of 0.5 each.

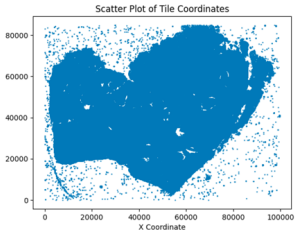
5.3 Filtering Out Background Tiles



(a) Example WSI Image



(b) Remaining tile distribution after filter with std threshold of 3



(c) Remaining tile distribution after filter with std threshold of 12.5

Figure 5: Example Resulting Distribution of Tiles in WSI Images after Standard Deviation Filter.

Within the Whole Slide Images, the areas surrounding the scanned tissue is often occupied by white background. From the perspective of building a model to segment regions of cells within the tissue, this background does not serve any useful purpose and as such filtering it out of the WSIs was determined to be a critical step in reducing noisy data that does not have a function within the created dataset.

As such, the standard deviation of each generated tiles' RGB values were calculated, and an arbitrary filter discarded tiles with standard deviation below some threshold. After some experimentation, it was determined that 12.5 was ideal for retaining the tissue mass within the WSI whilst removing the background tiles.

5.4 Greyscale images

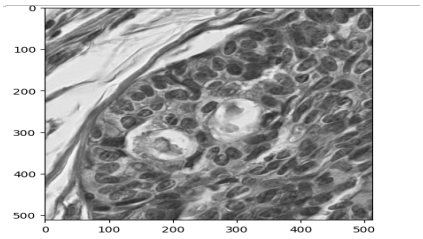


Figure 6: Example Greyscale Tile

The use of greyscale images is an technique for the processing and analysis of histological data. The process of converting color images to greyscale involves reducing the three channels of color information (red, green, and blue) in to a single channel. This transformation is to emphasize the texture and shape of the tissue structures without the distraction of color variations.

The conversion process from RGB to greyscale is to reduce computational complexity and focuses the analysis on structural details, which can vary due to staining differences or lighting conditions [15]. RGB images are converted to greyscale using a weighted sum approach, where the RGB channels are combined using specific weights.

5.5 Normalization of Tiles

A common technique in data normalization for image-based deep learning with RGB images is to normalize values to preset means and standard deviations. One such set of values is that of ImageNet [16], which has been trained on multitudes of images and thus forms a solid basis for modifying our dataset. The generated tiles from the slide were passed through this normalization prior to being inputted onto a model.

6 Training

6.1 Data structure

Due to the large image size, it was identified that training on multiple images was not feasible on ordinary computers. Therefore, we decided to start with one Whole Slide Image (WSI), BRAC_1494, which contains 36 UDH regions. Tiles of size 512 by 512 pixels were generated with no overlaps, resulting in 17,552 tiles, of which 379 contained masks. The data was only feasible to run on Google Colab (NVIDIA T4 GPU) with a batch size of 8 for grayscale tiles and a batch size of 4 for RGB tiles. In the final two weeks of the project, we obtained access to Bunya, the High Performance Computer at UQ, which features an A100 NVIDIA GPU node with 80 GB GPU RAM. With this enhanced computational power, we were able to utilize three WSIs (BRAC_1494, BRAC_1496 and BRAC_1286) with 49616 tiles and 743 tiles with masks and batch size of 16.

Due to the sparsity of the mask tiles and large amount of tiles, we employed a 1:3 ratios of mask tiles with non-mask tiles to obtain our dataset that used all the mask tiles and randomly chosen non-mask tiles. Then a 8:2 splits between the training and the testing set was employed.

6.2 Loss function and Optimizer

We employed the Binary Cross-Entropy with Logits Loss (BCEWithLogitsLoss), which is ideal for binary classification tasks such as segmenting medical images where pixels are categorized as masked or unmasked. This loss function is numerically stable and integrates a sigmoid activation with the BCE loss, making it efficient for our needs. The Adam optimizer was chosen for its robustness in handling sparse gradients and noisy data typical of medical images.

6.3 Tuning Hyperparameters

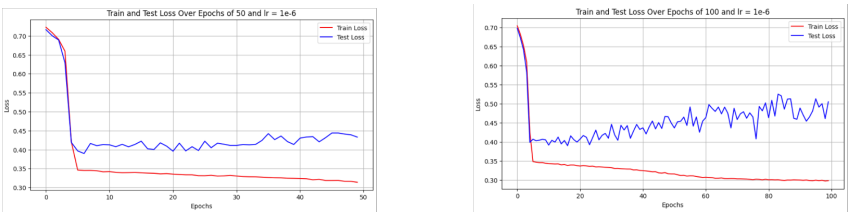
Through experimental analysis, we trained the vanilla U-Net using greyscale image, 1 WSI and 3 WSIs dataset with learning rate = 10^{-2}, 10^{-4}, 10^{-5}, 10^{-6}. The model performed poorly (see section 7.1, 7.1.1) and appendix A and B. Hence, we modified the model with data augmentation and ResNet as an encoder as described in section 4.1 and section 4.2.

7 Findings

7.1 Vanilla UNet

The Vanilla UNet model was trained on grey scale, 1 WSI and 3 WSIs data. However, the grey scale and 1 slide performance was extremely poor. The loss curve did not converge below 0.5 for both the training and the testing data. Therefore, in the following section we only report the findings for the 3 WSIs.

7.1.1 3 WSIs findings



(a) UNet with learning rate of 10^{-6} and 50 epochs

(b) UNet with learning rate of 10^{-6} and 100 epochs

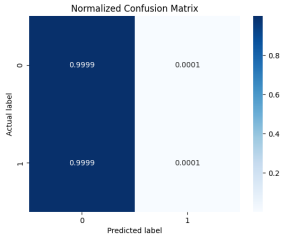
Figure 7: Training and Testing Loss Curve for Vanilla UNet, The plots shows that learning rate of 10^{-6}. The training set converged as the epoches increased. The test set for 50 epochs converged below 0.4 and the testing set for 100 epochs fluctuates around 0.45.

The 3 WSIs data were trained with the learning rate of 10^{-2}, 10^{-4}, 10^{-5} and 10^{-6}. For learning rate greater than 10^{-6}, the loss curve for the testing set did not converge as show in figure 15 in

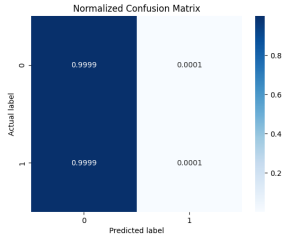
Appendix A. For learning rate of 10^{-6}, the loss curve converged better for the testing set as show in Figure 12a and 12b. Especially, the test loss is below 0.40 for 50 epochs. After the model is trained, we computed the confusion matrices for learning rate of 10^{-6} with 50 and 100 epochs on the training and testing sets. As shown in Figure 8. The model were able to picked up some masked signals as shown in the top left corner of each matrix. However, most of the masked regions are not identified. This demonstrated that the U-Net model performed poorly. The accuracy, precision, recall and F1 score were also calculated as shown in table 3. The high accuracy were obtained due to the large proportions of non-masked pixels compared to the masked pixels (i.e. the model was able to identified non-masked region well). The three other measurements showed that, the model was not able to pick up the masked regions which we desired to identified for this project.

The loss curves for the U-Net model trained at different learning rates and epochs show that learning Rate 10^{-6} with 50 and 100, both curves display a significant gap between the training and testing losses, with the gap becoming more pronounced at 100 epochs. This indicates potential overfitting where the model memorizes training data, leading to poor generalization on test data. Moreover, learning rates above 10^{-6} with 50 epochs results in the loss curve having higher fluctuations and overall higher loss values compared to the lower learning rate (as shown in Appendix A). This can indicate that the higher learning rate might be causing the training process to be less stable.

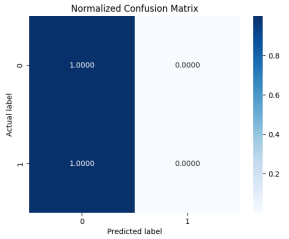
The normalized confusion matrices from the training and testing datasets at different epochs consistently. While the model is highly effective at identifying negative cases (as evidenced by the high true negative rates), its ability to recognize positive cases is quite poor (reflected in the high false negative rates and low true positive counts). The disparity between detecting negatives and positives may point to issues such as class imbalance.



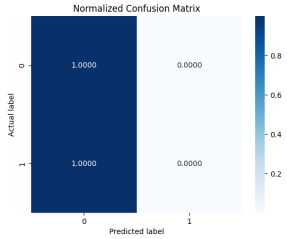
(a) UNet with learning rate of 10^{-6} on training data over 50 epochs



(b) UNet with learning rate of 10^{-6} on test data over 50 epochs



(c) UNet with learning rate of 10^{-6} on training data over 100 epochs



(d) UNet with learning rate of 10^{-6} on test data over 100 epochs

Figure 8: Comparison of UNet performance with learning rate of 10^{-6} across different epochs. The values are normalized row-wise by the total number of mask and non-mask pixels.

Metrics show high accuracy (above 0.87) contrasts sharply with extremely low precision and recall in detecting one of the classes (indicated by values close to zero for recall). This suggests an imbalance in the dataset where one class dominates or the model is biased towards one class and almost always predicts one class (likely the majority class), failing to detect the minority class effectively.

Table 1: U-Net Performance with Learning Rate of 10^{-6}

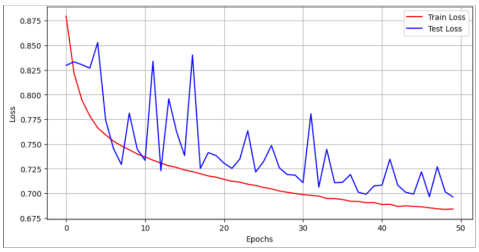
Epochs	Dataset	Accuracy	Precision	Recall	F1 Score
50	Train	0.8708	0.2277	0.0001	0.0002
50	Test	0.8756	0.2632	0.0001	0.0003
100	Train	0.8707	0.2230	0.0000	0.0000
100	Test	0.8766	0.2564	0.0000	0.0001

The model's training and testing performance indicate potential overfitting at lower learning rates as the epochs increase, with better convergence but poorer generalization to new data as shown in Appendix A.

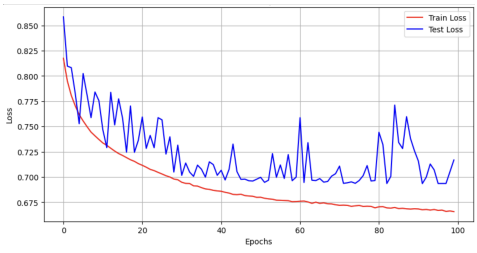
7.2 UNet with ResNet50 Encoder using SMP Package

The 3 WSIs were then trained on the SMP model [17] with epochs of 50 and 100 with the learning rate of 10^{-6}. Moreover, data augmentation with random flipping were added to the training dataset which increase the training dataset by 50 %.

The loss curves in figure 9 show the training and testing loss over epochs for a model trained with a learning rate of 10^{-6}. There's a visible gap between the training and testing loss, particularly in the 100 epochs graph in figure 9b. This could suggest some degree of overfitting, where the model performs better on the training data than on unseen test data. However, both curves trend downwards, which is positive and the model with 100 epochs seems to be closer to convergence, as indicated by the flattening of the loss curve, compared to the 50 epochs model.



(a) Training and Testing Loss Curve for SMP Model with learning rate of 10^{-6} and 50 Epochs

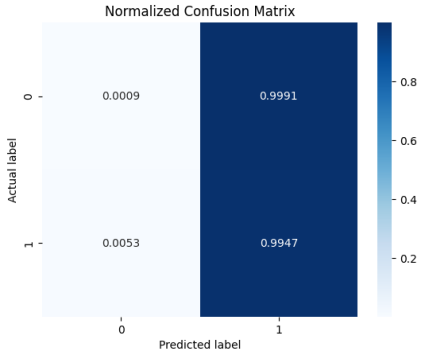


(b) Training and Testing Loss Curve for SMP Model with learning rate of 10^{-6} and 100 Epochs

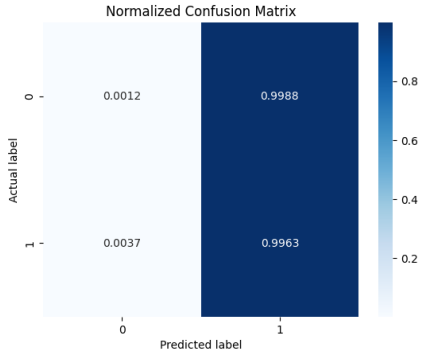
Figure 9: The Loss curve of the training and testing for SMP model

The confusion matrices in figure 10 for model training set and test set shows that both matrices have a very high true positive rate (masked), indicating that the model is highly effective at identifying the positive class. On the other hand, the extremely high false positive rates are concerning, as the model frequently misclassifies negative samples as positive. Therefore, the model is incorrectly labeling almost

all negative (non-masked) samples as positive (masked). This could lead to significant inefficiencies or issues, depending on the application of the model, as almost every non-masked sample is being flagged incorrectly.



(a) UNet + ResNet50 using SMP package on training data with learning rate of 10^{-6} and 100 epochs



(b) UNet + ResNet50 using SMP package on test data with learning rate of 10^{-6} and 100 epochs

Figure 10: Comparison of SMP performance with learning rate of 10^{-6} across different epochs. The values are normalised row-wise by the total number of mask and non-mask pixels.

These metrics indicate that while the model is adept at identifying positives (high recall), its practical effectiveness is limited due to a high rate of false positives (low precision) and an overall low accuracy, signaling a need for significant model adjustments or data rebalancing. Despite the convergence of the training and testing loss for 50 epochs, the model did not pick up any mask region as shown in the confusion matrix in Appendix C. This demonstrated that we required more epochs for the model to learn.

Table 2: U-Net + ResNet50 using SMP Package [17] Performance with Learning Rate of 10^{-6}

Epochs	Dataset	Accuracy	Precision	Recall	F1 Score
100	Train	0.1658	0.1653	0.9947	0.2834
100	Test	0.1772	0.1765	0.9963	0.2999

The SMP model, trained on three WSIs with up to 100 epochs and a learning rate of 10^{-6}, shows a discrepancy between training and testing loss, hinting at potential overfitting, particularly evident in the 100 epochs curve. Despite downward trends in the loss curves, suggesting improvement, the model struggles with specificity, as evidenced by high false positive rates in confusion matrices. This misclassification of almost all negative samples as positive severely undermines its practical utility. High recall rates contrast sharply with low precision and overall accuracy.

7.3 UNet with DenseNet121 Encoder

7.3.1 DenseNet121 Classifier

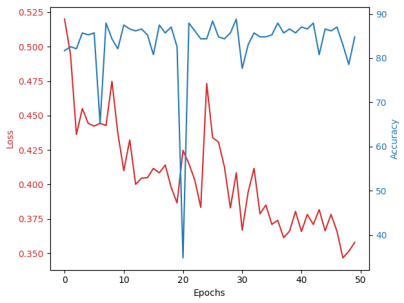


Figure 11: Graph of Training Loss and Validation Accuracy during Training of DenseNet

Initially, a DenseNet121 model [18] was constructed and trained on the tiled images as a binary classification model, outputting whether a tile contained any UDH regions. Similar to the segmentation models discussed in this report, the training dataset used oversampling of the minority UDH tiles such that there was an even ratio of the two classes, although in the larger dataset generated from the tiles the ratio of UDH to non-UDH regions was about $45:1$.

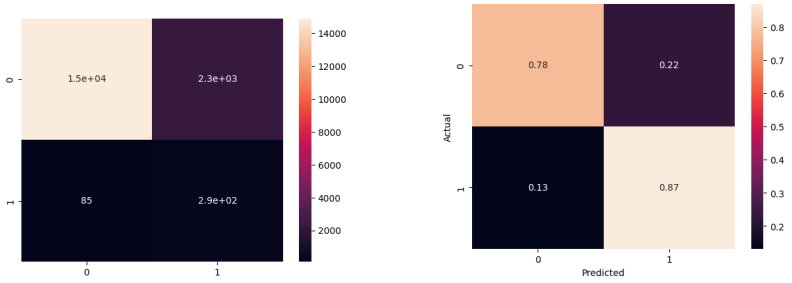
Cross entropy loss was utilized in training due to its logarithmic property of punishing confidently incorrect outputs from a model [19]. Different learning rates were examined however 10^{-6} proved similarly effective in training the DenseNet as it did training the segmentation models. During training, high accuracy in the validation set was immediate after the first epoch with minimal fluctuation towards improvement. This could suggest that the classifier's confidence in outputs grew during training, however what it considered to be the more likely option of the two classes remained relatively unchanged.

Table 3: Metrics of DenseNet Classifier on Test Set

Accuracy	Precision	Recall	F1 Score
0.8667	0.5547	0.8219	0.5639

As shown in table 3, when tested on the entire dataset generated from the tiles, the model was able to achieve roughly 87% accuracy, however a relatively low precision of 55% occurred due to overestimation by the model of the frequency of UDH regions in the far larger non-UDH majority class.

The normalized confusion matrix in Figure 12b suggests that the model correctly identified the non-UDH regions as such for 78% of instances, however the massive disparity between the two class types infers that precision would suffer regardless.



(a) Raw Confusion Matrix **(b)** Normalized Confusion Matrix

Figure 12: Raw and Normalized Confusion Matrices for DenseNet Classifier, 1 is UDH regions and 0 is non-UDH.

7.3.2 DenseNet121-Unet Integration

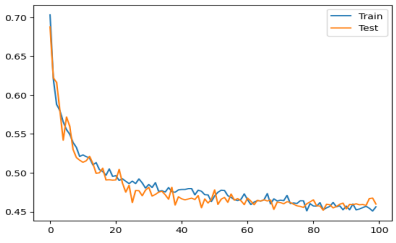


Figure 13: Graph of Average Epoch Loss on Training and Test Set for UNet with DenseNet Encoder

The weights of this DenseNet model were then transferred to a UNet encapsulating the DenseNet as an encoder. The model was then trained with the previously most successful learning rate of 10^{-6} for 50 epochs, wherein the output suggests that the training and test loss were far closer connected with this new encoder. However, the resulting loss values were still very distant from ideal. The model failed to conform to the mask pixels, and instead akin to other models have been practiced, it plateaued once it began essentially outputting zero matrices. This is a recurrent theme within the trained models, wherein the underlying patterns and relatively sparse samples have made it such that loss is minimized by identifying no areas to segment.

8 Limitation

8.1 Limited computational resource

Initially, our project utilized an Intel Core i5 13500HX with an NVIDIA GTX 4050 GPU and 24 GB of system RAM. This setup proved inadequate for training a U-Net model on large-scale image data, as training times exceeded one hour per epoch, making it impractical.

To address this, we transitioned to Google Colab Pro, utilizing a more powerful NVIDIA Tesla T4 GPU. This significantly improved our efficiency, reducing the training time for our RGB model to about one minute per epoch with a 256 x 256 image input size and a learning rate of 1e-4.

Parallel attempts to train a greyscale model on a MacBook Pro with an M1 chip showed that a 128x128 input size was still infeasible with over 50 minutes per epoch. However, reducing the image size to 64x64 and increasing the learning rate to 1e-3 decreased the epoch time to under three minutes.

We reliance on Google Colab Pro to train for our 1 WSI, which balanced computational power with accessibility, allowing our project to proceed efficiently despite initial hardware limitations. Then in the final 2 weeks one member of the team gained access to Bunya where it is feasible to run larger dataset but time was a limiting factor as training for 50 epochs for batch size of 16 took about 2 hours.

9 Discussion and Conclusion

The original aim of identifying UDH region in the breast tissue H&E images were not achieved. U-Net performance were poor and only small proportion of UDH regions were identified. The high accuracy score of 87 were due to large amount of non-mask pixels presented. Nevertheless, the model performed poorly as it was not able to identified masked regions. The model with the learning rate greater than 10^{-6} did not pick up the masked signal at all as shown in Appendix B. This could be due to the sparsity of the masked pixels, optimal hyperparameters were not found or the RGB images did not provide enough information about the tissue. Unexpectedly, with the more complex ResNet50 Encoder as encoder with tile augmentation. The model were able to identify the masked images better compared to the non-masked with the high recall scores of 99. However, the low precision scores showed that the model over fit to the masked signals. Despite the limitations described in the section above, U-Net or modified versions of U-Net could still be a good model for cell segmentation tasks. Glänzer et al

(2023) [20] used the unsupervised adaptation technique with U-Net with ResNet structure as encoder shown great performance in segmenting the sparse regions in the blood vessels.

The model can also be improved by incorporating spatial transcriptomics data. The genetic expression levels for different cells are different. The advance in technologies allow us to spatially locate these gene expressions. Hence, identifying different cell types at different location that can potentially improve the segmentation tasks. This could be interesting area for further explorations [21].

The metrics from the DenseNet classifier proved that there are likely to be identifiable patterns within the dataset that can be resolved to UDH regions, however it is possible that segmentation of these regions required a larger dataset than what was available to train on given the computational resources.

In conclusion, the U-Net and SMP model performed poorly on the segmentation of the UDH regions of the 3 WSIs of the BRACS dataset. The DenseNet classifier showed that it was possible to identified whether a tile has UDH region showing that the features from the H&E images is learnable and used. Further explorations and modifications on U-Net is needed to potentially achieve good performances on the segmentatioin task.

References

[1] N. Brancati, A. M. Anniciello, P. Pati, *et al.*, "BRACS: A dataset for BReAst carcinoma subtyping in H&E histology images," en, *Database (Oxford)*, vol. 2022, Oct. 2022.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Lecture Notes in Computer Science*, ser. Lecture notes in computer science, Cham: Springer International Publishing, 2015, pp. 234–241.

[3] A. Ben Hamida, M. Devanne, J. Weber, *et al.*, "Deep learning for colon cancer histopathological images analysis," en, *Comput. Biol. Med.*, vol. 136, no. 104730, p. 104 730, Sep. 2021.

[4] Y. Li, N. Li, X. Yu, *et al.*, "Hematoxylin and eosin staining of intact tissues via delipidation and ultrasound," en, *Sci. Rep.*, vol. 8, no. 1, p. 12 259, Aug. 2018.

[5] J. de Matos, S. Ataky, A. de Souza Britto, L. Soares de Oliveira, and A. Lameiras Koerich, "Machine learning methods for histopathological image analysis: A review," en, *Electronics (Basel)*, vol. 10, no. 5, p. 562, Feb. 2021.

[6] S. M. Patil, L. Tong, and M. D. Wang, "Generating region of interests for invasive breast cancer in histopathological whole-slide-image," in *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, Madrid, Spain: IEEE, Jul. 2020.

[7] N. Dong, M. Kampffmeyer, X. Liang, Z. Wang, W. Dai, and E. P. Xing, "Reinforced Auto-Zoom net: Towards accurate and fast breast cancer segmentation in whole-slide images," 2018.

[8] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-net and its variants for medical image segmentation: A review of theory and applications," *IEEE Access*, vol. 9, pp. 82 031–82 057, 2021. DOI: 10.1109/ACCESS.2021.3086020.

[9] W. Baccouch, S. Oueslati, B. Solaiman, and S. Labidi, "A comparative study of cnn and u-net performance for automatic segmentation of medical images: Application to cardiac mri," *Procedia Computer Science*, vol. 219, 2023, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2023.01.388. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050923003976.

[10] L. Ali, F. Alnajjar, H. Jassmi, M. Gochoo, W. Khan, and M. Serhani, "Performance evaluation of deep cnn-based crack detection and localization techniques for concrete structures," *Sensors*, vol. 21, p. 1688, Mar. 2021. DOI: 10.3390/s21051688.

[11] C. Zhang, P. Benz, D. M. Argaw, *et al.*, "Resnet or densenet? introducing dense shortcuts to resnet," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2021, pp. 3550–3559.

[12] L. Shi, H. Xue, C. Meng, Y. Gao, and L. Wei, "DSC-OpenPose: A fall detection algorithm based on posture estimation model," in *Lecture Notes in Computer Science*, ser. Lecture notes in computer science, Singapore: Springer Nature Singapore, 2023, pp. 263–276.

[13] N. Aldoj, F. Biavati, F. Michallek, S. Stober, and M. Dewey, "Automatic prostate and prostate zones segmentation of magnetic resonance images using DenseNet-like u-net," en, *Sci. Rep.*, vol. 10, no. 1, p. 14 315, Aug. 2020.

[14] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," 2017.

[15] Y. Ge, Q. Zhang, Y. Sun, Y. Shen, and X. Wang, "Grayscale medical image segmentation method based on 2d&3d object detection with deep learning," *BMC Medical Imaging*, vol. 22, no. 1, p. 33, 2022, ISSN: 1471-2342. DOI: 10.1186/s12880-022-00760-2. [Online]. Available: https://doi.org/10.1186/s12880-022-00760-2.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[17] P. Iakubovskii, *Segmentation models*, https://github.com/qubvel/segmentation_models, 2019.

[18] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, *Densely connected convolutional networks*, 2016. eprint: arXiv:1608.06993.

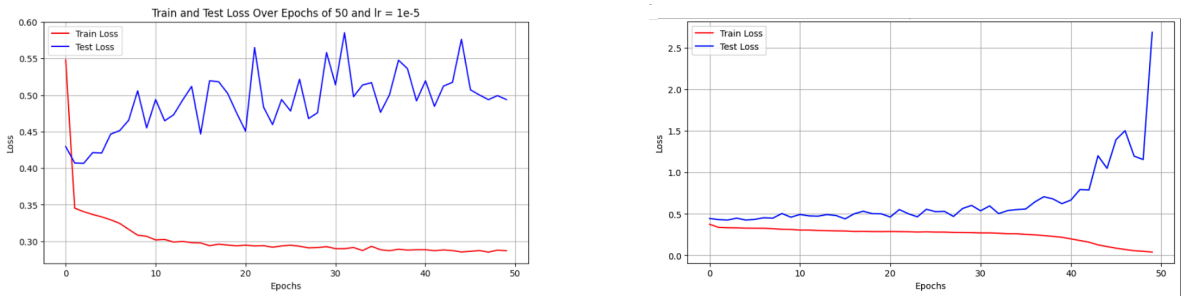
[19] A. Mao, M. Mohri, and Y. Zhong, "Cross-entropy loss functions: Theoretical analysis and applications," in *Proceedings of the 40th International Conference on Machine Learning*, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., ser. Proceedings of Machine Learning Research, vol. 202, PMLR, 23–29 Jul 2023, pp. 23 803–23 828. [Online]. Available: https://proceedings.mlr.press/v202/mao23b.html.

[20] L. Glänzer, H. E. Masalkhi, A. A. Roeth, T. Schmitz-Rode, and I. Slabu, "Vessel delineation using U-Net: A sparse labeled deep learning approach for semantic segmentation of histological images," en, *Cancers (Basel)*, vol. 15, no. 15, p. 3773, Jul. 2023.

[21] L. Moses and L. Pachter, "Museum of spatial transcriptomics," en, *Nat. Methods*, vol. 19, no. 5, pp. 534–546, May 2022.

Appendix

Appendix A: Training and Testing Loss Curves for U-Net Models with learning rate: 10^{-2}, 10^{-4} and 10^{-5}



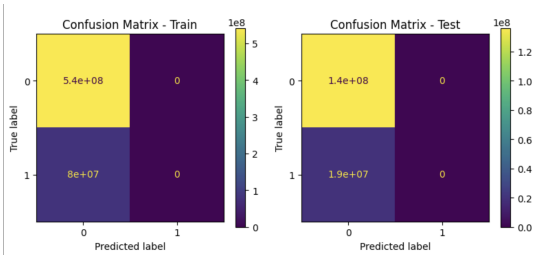
(a) UNet with learning rate of 10^{-5} and 50 epochs

(b) UNet with learning rate of 10^{-4} and 50 epochs

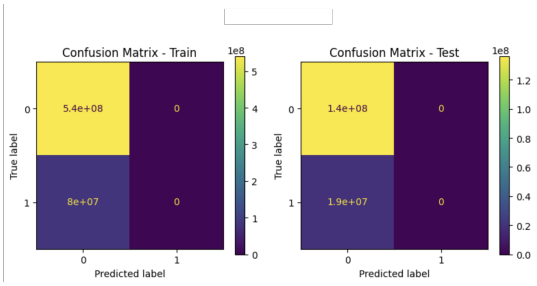
Figure 14: Training and Testing Loss Curves for Vanilla UNet. (a) shows the loss curves for learning rate of 10^{-5} (b) shows the loss coruves for learning rate of 10^{-4}. The testing set diverges and for 10^{-4} and the testing set did not converge below 0.4.

Note: the learning rate of 10^{-2} is not reported because it was clearly not converging during the training process. Hence, interrupted at the earlier epoch.

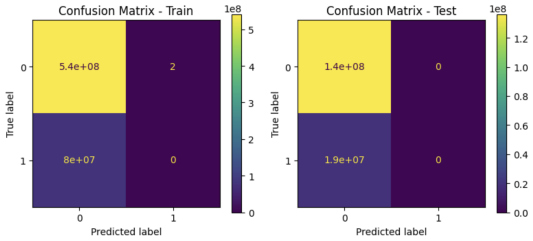
Appendix B: Confusion Matrix for U-Net Models with learning rate: 10^{-2}, 10^{-4} and 10^{-5}



(a) raw confusion matrix for training and testing for the learning rate of 10^{-2}



(b) raw confusion matrix for training and testing for the learning rate of 10^{-4}



(c) raw confusion matrix for training and testing for the learning rate of 10^{-5}

Figure 15: Confusion matrices for the training and testing for the U-Net model, showing that the model did not pick up the masked pixels at all

Appendix C: The Raw Confusion Matrix for 50 epochs of SMP model with Learning Rate of 10^{-6}

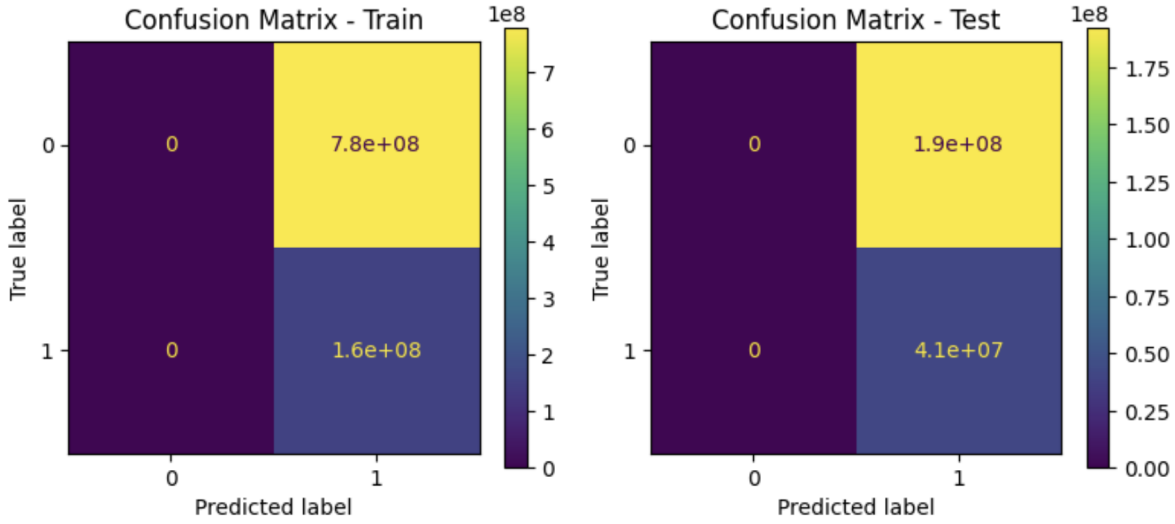


Figure 16: Loss curve for SMP model for over 50 epochs with learning rate of 10^{-6}. Although the loss curves were converging, it was cleared that we required more epochs