Journal Title Here Vol. , No. , pp. 1–6
doi: DOI HERE
Advance Access Publication Date: Day Month Year
Paper

Image Segmentation for Cell Image Datasets

Ashton Frias[1,*]

*Ashton Frias. as700045@ucf.edu

ABSTRACT

Image segmentation models reduce the need for manual labeling, accelerating research and advancement in bioinformatics. However, challenges such as smaller dataset sizes and model architecture constraints have potential for improvement. Implementing different techniques on the datasets and model architecture has potential to improve performance. In this paper, I test the effects of training the two models Unet and Swin-Unet on various sized datasets, while implementing different augmentation techniques. In addition, I also modified the skip connection layers in both models to see if it would enhance their segmentation performance. Experimental results show that modifying the skip connection layers matches or outperforms the standard architecture by approximately 1% for top-performing models on each dataset.

1.1 INTRODUCTION

In bioinformatics, many scenarios in research require distinguishing the structures of a cell. This can range from recognizing one to hundreds of structures in a single image. As a result, manually labeling each part of the cell is tedious and prone to human error. Researchers utilize semantic segmentation models to alleviate this problem. These models assign a class to each pixel that results in the segmented image (2).

However, creating these datasets to train these segmentation models takes time and often results in either small datasets with quality labels or large datasets with subpar labels. Segmentation models are often plagued by overfitting and generalization due to the small datasets or poor segmentations due to subpar labels. To address the issue of small datasets, researchers employ techniques such as data augmentation that increase the size of the dataset artificially. Additionally, modifying the architecture of the model can also increase performance. For example, replacing the standard skip connection layers in UNet with a fusion layer increased performance (4). By utilizing these techniques, researchers can develop a robust segmentation model that accurately identifies the structure of a cell without compromising performance.

In this paper, I research the effects of training two segmentation models, UNet and Swin-UNet (1) (2), across three datasets: BCSS, BCSS512, and MoNuSeg (7) (6). Next, I modified the skip connection layers in both models to determine whether these modifications would enhance the models' performance and further reduce the semantic gap between the encoder and decoder compared to its standard architecture. The first modification involved adding a convolutional layer into the standard skip connection layer (9). The second modification introduced a fusion skip connection layer that fuses all the feature maps from each encoding layer (4). As a result, the modified skip connection layers maintained the same level of accuracy or improved it by approximately 1%.

1.2 Related Works

Recently, encoder-decoder architectures have become a popular choice for segmentation tasks because they are effective at capturing spatial and semantic information in images (2). Models like U-Net are a perfect example of this, as the encoder is responsible for extracting feature maps from every level of its encoding layer, while the decoder maps those features to a segmentation. Other models, such as Swin-Unet, build upon this by using a vision transformer encoder that leverages a shifted windowing scheme (1), that captures local features and global features that are otherwise ignored in models like U-Net. Both models can further be improved by either modifying the model's internal architecture or by implementing an ensemble/double-architecture as seen in DoubleU-Net (5). Because of added computational complexities when implementing an ensemble/double-architectures, I opted for modifying the internal architecture.

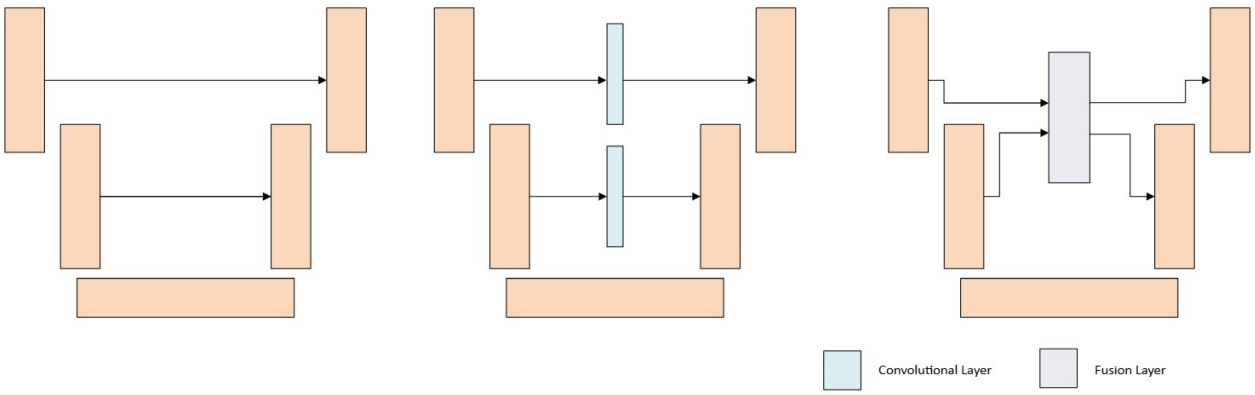

Fig. 1. An overview of the model architectures with modified skip connectors used in this paper. The details for the encoder and decoder architectures can be found in the respective source papers (1) (2).

.

U-Nett++ modifies the skip connection layers by employing nested skip pathways to reduce the semantic gap further when compared to the traditional U-Net Model (8). Another segmentation model that modifies the skip connectors is called FusionU-Net, which fuses all of the feature maps through a fusion module to reduce semantic gaps (4). By combining low and high-level features from each encoding layer, it can provide more information to each decoding layer. Both approaches enhance the effectiveness of the traditional U-Net model while maintaining a low computational cost. In this paper, I research the effects of modifying the skip connection layers in both U-Net and Swin-Unet, by adding a convolutional layer or a fusion layer, and evaluate their performance while training on different-sized datasets.

2.1 METHODS

2.2 Architecture

For UNet and Swin-Unet, I followed the basic encoder and decoder architecture for both models provided in their respective source papers (1) (2) with a few changes. In terms of input size, I reduced it to 224×224×3 from the original 572×572×3, to decrease the computational complexity. For Swin-Unet, since training a vision transformer from scratch requires a significant amount of data to train effectively, I opted for a pre-trained encoder provided by the original paper's authors (3). With this, I could train effectively on smaller datasets, which would not be possible with an untrained Swin transformer encoder.

For selecting the appropriate loss function for both models, it was dependent on the dataset. When training on MoNuSeg, which consisted of 1 class label, the Binary Cross-Entropy loss function was used because it is designed for binary classification. This is because, in MoNuSeg, the models task is predict whether a pixel belongs to a nucleus.

For BCSS and BCSS512, these datasets contain multiple class labels, which require two loss functions. The first loss function selected was Cross Entropy Loss to predict which class a pixel belongs to. The second loss function was Dice's Loss to measure the accuracy and overlap of the predicted segmentation to the ground truth. When combined, these make the models more robust and less susceptible to class imbalance.

Next, I wanted to investigate whether modifying the skip connection layers would enhance the accuracy as observed in other implementations, such as FusionU-Net (4). Therefore, I made two additional modifications - a convolutional and fusion skip connection layer as shown in Figure 1. The goal was to see if this would reduce the amount of information lost as the input transitioned from the encoder to the decoder (4).

The architecture for the convolutional skip connection layer consisted of obtaining the feature map from the encoder and passing it through a 1×1 convolutional layer before concatenating it with the decoder. This additional layer within the skip connection can help suppress noisy channels while amplifying valuable ones (9).

For the fusion layer, instead of passing individual feature maps from the encoder to each of the corresponding decoding layers, all of the feature maps from each encoding layer are fused. To achieve this, all feature maps that are not produced on the current encoding layer are passed through a 1×1 convolutional layer to match the number of channels, and then it is resized using bilinear interpolation to match the resolution. Next, all of the feature maps are fused using element-wise addition, and then sent through another 1×1 convolutional layer before concatenation with the decoder's feature map. This concept is similar to the architecture of FusionU-Net (4). By providing the corresponding decoding level with the fusion of all encoded feature maps, the model has more context to make a more

accurate prediction. The goal of this approach is to see if it could further enhance the information flow and reduce the semantic gap between the encoder and decoder (2).

Model	**Parameter Size**
UNet	31,043,521
Swin-UNet	35,383,483

Table 1. Comparison of parameter sizes between UNet and SwinUNet

2.3 Dataset

For the datasets used, I selected three of varying sizes (small, medium, and large). All of these datasets consist of paired images and pixel-level segmentation masks.

The first dataset used was MoNuSeg (Multi-organ nucleus Segmentation), which contains carefully annotated images of tumor nuclei in different organs (6). The objective of this dataset is for the segmentation model to identify pixels that belong to tumor nuclei. Training models on this dataset is important because they can not only help researchers identify the cancer grade but also predict the effectiveness of the patient's current treatment plan (6). The original images in MoNuSeg had a resolution of 1000×1000×3 pixels. Since the models only accept 224×224×3, I resized the images to 896×896×3, allowing me to divide them evenly without the need for padding. This resulted in 704 image and mask pairs. I used 70% for training, 20% for validation, and 10% for testing.

The next two datasets are both a part of the BCSS (Breast cancer semantic segmentation) challenge and will be used for my medium and large dataset evaluation (7). These datasets were part of an experiment to explore whether crowd-sourcing is a feasible option for creating large-scale cell segmentation datasets. This process consisted of a multi-stage pipeline. In the first stage, all of the images were processed by 20 non-pathologists. In the second stage, their work was refined by three junior pathologists, and finally, in the last stage, it was refined again by two senior pathologists (7). While the dataset was large, some of the segmentations lacked the precision and accuracy typically found in smaller datasets, where pathologists can spend more time carefully segmenting each part of the image with pixel-level accuracy. This level of detail is otherwise missed when creating large datasets. Despite this, BCSS is valuable for assessing whether quantity can compensate for quality in training an effective segmentation model.

The first dataset in this challenge was BCSS, which had an image resolution of 224×224×3 with three class labels. The training set had 30,760 images, and 4,021 images for the test set. Since the image resolution size matched the input for both of the proposed models, there was no need for augmentation. The purpose of this dataset is for the model to identify cancerous tissues.

The second dataset in this challenge is BCSS512, a subset of BCSS, which consists of more detailed segmentation. This dataset comprises 22 class labels, representing various biological structures in breast tissue, including tumors, stroma, fat, and plasma cells. The dataset was divided into 5,400 images and mask pairs for training, 2,768 for validation, and 600 for testing. Additionally, since the images and masks did not match the input size for the proposed models, each sample was resized to 224×224.

2.4 Augmentation

For augmentation, I applied commonly used techniques to each image during training to increase the size of the datasets artificially. The first three augmentation techniques I used were horizontal/vertical flips and rotations. This is designed to help the model make more accurate predictions, regardless of the object's orientation. The last three techniques were elastic transformation, grid distortion, and brightness changes, all of which help improve the models' generalization to unseen data.

2.5 Metrics

The metrics used to test the effectiveness of the models were Mean Intersection over Union (mIoU) and F1 score.

Equation 1 represents the formula for Mean Intersection over Union (mIoU), where C is the total number of classes and i is the current class. The terms P_i and G_i represent the predicted segmentation and the ground-truth segmentation. The intersection $|P_i \cap G_i|$ represents the total number of correctly predicted pixels, and the union $|P_i \cup G_i|$ stands for the total number of predicted pixels for that class. This equation evaluates how well the model's predicted segmentation matches the ground truth.

$$\text{mIoU} = \frac{1}{C} \sum_{i=1}^{C} \frac{|P_i \cap G_i|}{|P_i \cup G_i|} \tag{1}$$

Equation 2 represents the F1 score. This measures a model's performance on an imbalanced dataset, a common occurrence in segmentation datasets.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2}$$

Both metrics range from 0-1, with 1 indicating that all predictions perfectly align with the ground truth.

3.1 ANALYSIS AND RESULTS

3.2 Training

I implemented these models using PyTorch and trained them using NVIDIA 3070.

When training on MoNuSeg, both models were trained for 50 epochs with a batch size of 8. Training took around

1 hour for each Swin-UNet combination and 2 hours for each UNet combination. Furthermore, I used the Adam optimizer with a learning rate of 10e-3. As shown in Figure 2, the training and validation losses stabilized around 25 epochs for UNet, a trend consistent across all model combinations I trained.

For BCSS and BCSS512, both models were trained for 10 epochs with a batch size of 50. However, due to the added computational complexity of the fusion skip connection layers for UNet combined with larger datasets, I could not train it because the available GPU resources were insufficient. For training times, BCSS training took approximately 50 hours for each U-Net combination and 10 hours for each Swin-Unet combination. For BCSS12, training took around 10 hours for Unet and 2 hours for each Swin-Unet combination. The vast differences in training times can be attributed to the use of a pre-trained encoder for Swin-Unet. For both datasets, I used the Adam optimizer, but I used different learning rates for the models, 10e-3 for Unet and 10e-5 for Swin-Unet. The lower learning rate was necessary for Swin-Unet convergence because if it was equivalent to UNet, it resulted in random guessing. For BCSS and BCSS512, as shown in Figure 2, the training and validation losses have not yet stabilized, meaning that all models would benefit from additional training. Ideally, hyperparameter tuning would increase the performance of all models, however, due to the amount of time required to train all of the models, I was unable to implement this within the project timeline.

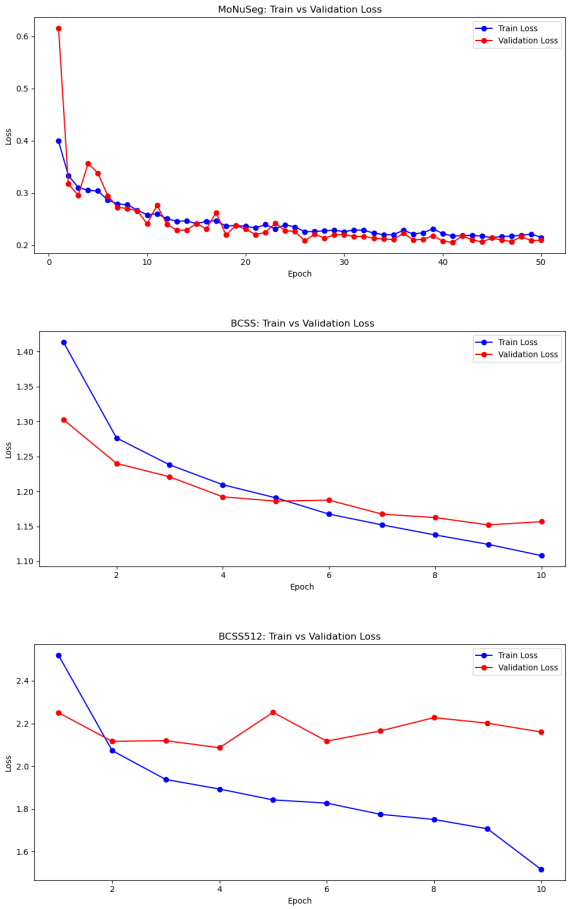

Fig. 2. Training and validation performance for each epoch: UNet Convolutional Network on the MoNuSeg dataset and Swin Convolutional Network on both BCSS datasets.

3.2 Results

When comparing the results shown in Table 2 between both UNet and Swin-Unet it is clear which model is more suited for different scenarios. In binary classification tasks, UNet's performance is better, likely because this architecture is more appropriate for low-complexity segmentation. With 4.3 million fewer parameters than Swin-UNet, as shown in Figure 1, UNet is less likely to overfit the training data. However, as the classes and complexity increase, Swin-UNet displays a significant advantage. This can be attributed to its powerful encoder, which extracts better global and local features, whereas UNet struggles with accomplishing this. We can also observe that the modified skip connection layers' performance matches or outperforms that of the standard architecture.

When comparing the performance of each combination for both models on the MoNuSeg dataset, we observe some improvements, most notably for the UNet model. Its convolutional skip connection layer outperforms the other layer by approximately 1-2%. This is impressive because it is less computationally intensive than the fusion skip connection layer. Its advantage likely stems from its ability to suppress noisy feature channels while amplifying useful ones.

For SwinUnet, a noteworthy aspect to highlight is that even though it performed worse than UNet, it required less training time because of its pre-trained encoder, accentuating the trade-off between efficiency and accuracy. Similar to UNet, the convolutional skip connection layer also performed the best. When comparing the output for the top-performing models in Figure 3 to the ground truth, we can see that both models effectively capture all of the segmented nuclei with some false positive segmentations. Overall, the performance of both models is impressive, however, there is potential for growth due to the amount of the false positives. For training, it seems that the loss leveled out around 25 epochs for both training and validation, meaning that the models potentially overfitted to the training data. Tweaks to the training duration of both models could benefit performance.

For both of the BCSS datasets, there is a significant difference between performances. In Table 2, Swin-Unet surpasses Unet by approximately 25% across all combinations, indicating that the attention mechanism

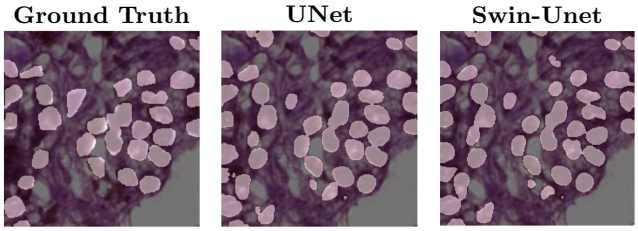

Fig. 3. Comparison of Ground Truth, Unet (Convolutional), and Swin-Unet (Convolutional) on the MoNuSeg dataset.

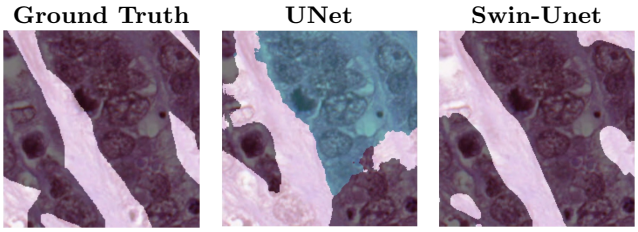

Fig. 5. Comparison of Ground Truth, Unet, and Swin Unet on the BCSS dataset.

within the Swin encoder can capture local and global features more efficiently as the complexity of the dataset increases. When comparing the outputs for the top-performing models as seen in Figure 4 and Figure 5, it is even more evident that UNet struggles to classify a majority of pixels and segmentations. On the other hand, Swin-UNet predictions are closer to the ground truth. Also, mirroring findings in MoNuSeg analysis, Swin-Unet trained more efficiently than Unet, only taking 10 hours to train compared to 50 hours. In addition, consistent with the results in MoNuSeg, the modified skip connection layers match or outperform the standard architecture across both datasets.

For BCSS, as seen in Table 2, the standard skip connection layer performed the best in terms of mIoU and F1 score. However, the scores of each of the connection layers remained similar. This was not the case for BCSS512, as the fusion skip connection layer performed approximately 1-3% better than the other two layers. One possible explanation is that for lower-difficulty tasks, such as MoNuSeg and BCSS, the semantic gap between the encoder and decoder is small. As a result, the additional information injected from all layers into a single feature map does not provide any advantage over the standard skip connection. However, as the difficulty increases, the wider the semantic gap becomes, and more information is lost. In this scenario, the fusion skip connection layer would provide more information, reducing the gap and recovering the necessary information for an accurate prediction.

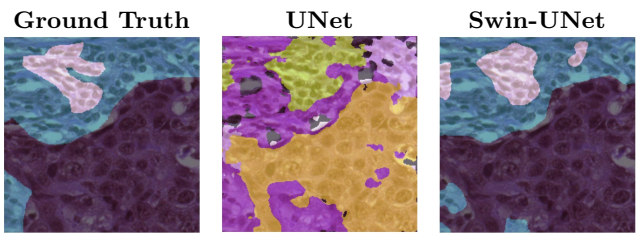

Fig. 4. Comparison of Ground Truth, Unet (Convolutional), and Swin Unet (Fusion) on the BCSS512 dataset.

4 CONCLUSION

In this project, I investigated the impact of modifying the skip connection layers in UNet and Swin-Unet. The goal was to achieve better results than the standard skipping connection layers, similar to FusionU-Net (4). My experiments provide evidence that modifying the skip connection layers can either match or outperform the standard architecture. However, it also shows that using modified skip connection layers may be situational, as in low-difficulty tasks where less information is required from the corresponding encoder to make an accurate prediction. For example, in MoNuSeg and BCSS, the simpler skip connections outperformed the Fusion layer. However, as the task difficulty increases, the potential for information loss between the encoder and decoder becomes higher. To remedy this, fused feature maps from every encoding layer can recover the lost information. This is shown in the result for BCSS512, the most challenging dataset in this project due to its class imbalance and larger number of classes. These findings highlight that while skip connection modifications can offer performance gains, their effectiveness may be situational and depend heavily on the complexity of the task.

4.1 Future Work

My work highlights the potential to increase the performance of UNet structured segmentation models by modifying the skip connection layers. Future work to further this research would include increasing the convolutional layers between the skip layers to see if it would benefit the model, as more layers could help extract or refine different features. Additionally, experimenting with the hyperparameters to observe if it can optimize these architectures further, specifically testing different loss functions, learning rates, and batch sizes. Lastly, conducting an in-depth analysis to determine at what point the model would benefit from using the fusion skip connection layer to reduce data loss between the encoder and decoder with increasing dataset complexity.

		BCSS		BCSS512		MoNuSeg	
Models	Skip Connection	**mIoU**	**F1**	**mIoU**	**F1**	**mIoU**	**F1**
U-Net	Standard	30.44	53.70	23.96	34.04	66.27	79.45
	Convolutional	33.78	47.42	23.45	33.70	**67.32**	**80.28**
	Fusion	–	–	–	–	65.68	79.03
Swin U-Net	Standard	**57.55**	**64.39**	47.45	53.84	63.98	77.65
	Convolutional	57.23	64.16	45.78	53.12	64.78	78
	Fusion	56.59	63.69	**48.11**	**55.06**	63.42	76.96

Table 2. Performance evaluation of different skip connection modifications across three datasets using mIoU and F1 score. Higher values indicate better performance.

REFERENCES

1. Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation *arXiv arXiv:2105.05537*, 2021.
2. Olaf Ronneberger, Philipp Fischer, and Thomas Brox U-Net: Convolutional Networks for Biomedical Image Segmentation *arXiv arXiv:1505.04597*, 2015.
3. Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo Swin Transformer: Hierarchical Vision Transformer using Shifted Windows *arXiv arXiv:2103.14030*, 2021.
4. Zongyi Li, Hongbing Lyu, and Jun Wang FusionU-Net: U-Net with Enhanced Skip Connection for Pathology Image Segmentation *arXiv arXiv:2310.10951*, 2023.
5. Debesh et al. DoubleU-Net: A Deep Convolutional Neural Network for Medical Image Segmentation *arXiv arXiv:2006.04868*, 2020.
6. Kumar et al. A Multi-Organ Nucleus Segmentation Challenge. *IEEE Transactions on Medical Imaging*, 39(5):1380-1391, 2020.
7. Amgad et al. Structured crowdsourcing enables convolutional segmentation of histology images *Bioinformatics*, 35(18):3461–3467, 2019. Available at: `https://doi.org/10.1093/bioinformatics/btz083`.
8. Zongwei Zhou et al UNet++: A Nested U-Net Architecture for Medical Image Segmentation *arXiv arXiv:1807.10165*, 2018.
9. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.