NYCU Spr2025 AI Capstone: Final Project Report

Topic: Medical Image Analysis — Binary Segmentation of Physiology Images

Group ID: D5
Authors: 111550057 莊婷馨, 111550108 吳佳諭，111550113 謝詠晴
Demo Link: 113-2 AIC Final Project - Group D5
GitHub: https://github.com/chia-yuu/AIC-final-project/tree/main

Member	Contribution
111550057 莊婷馨	DCAN model training, Evaluation Metrics, Presentation
111550108 吳佳諭	Preprocessing, Base, UNET model training, Presentation
111550113 謝詠晴	Data Analysis and Visualization, Report, Presentation

1 Introduction

1.1 Motivation

We are interested in image segmentation but have not previously worked with medical imaging. This Kaggle competition provides a clear, task-specific challenge, offering a great opportunity to explore segmentation techniques and evaluation methods in a practical context.

1.2 Dataset

We used the GlaS dataset from GlaS@MICCAI'2015: Gland Segmentation contest, with 165 physiology images, 165 masks and a csv file with classification labels. It contains 85 training images and 80 testing images, all in bmp file format.

1.3 Goals

1) Explore deep learning models for medical image segmentation
2) Implement and compare U-Net and DCAN architectures
3) Evaluate model performance on the Gland Segmentation dataset
4) Apply data augmentation techniques and investigate the impact
5) Compare model accuracy and boundary quality using metrics such as Dice, IoU, HD, and ASSD
6) Visualize and analyze segmentation results
7) Provide qualitative and quantitative comparisons

2 Techniques

2.1 Models

2.1.1 U-Net

U-Net is a widely used image segmentation model with a symmetric encoder–decoder structure. The encoder captures context through downsampling, while the decoder restores spatial details via upsampling. Skip connections link corresponding layers to combine high-resolution features, improving accuracy and preserving fine details. U-Net works well with limited data and trains quickly due to its simple design, but it can struggle with class imbalance and low-quality images.

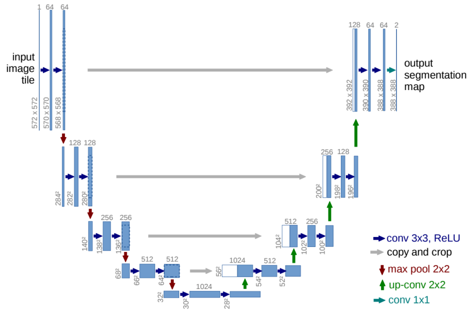

Fig 1. U-Net model architecture

2.1.2 DCAN

DCAN extends U-Net by introducing contour-awareness through a dual-decoder architecture. The encoder captures global context, while two separate decoders are used—one for learning the object mask and another for refining boundary details. This design improves boundary quality and is especially effective for tasks where precise edges are critical, such as medical image segmentation. It performs well on noisy or low-contrast data and helps distinguish closely attached structures, like clustered glands. However, the model is relatively large, leading to longer training times and higher computational costs.

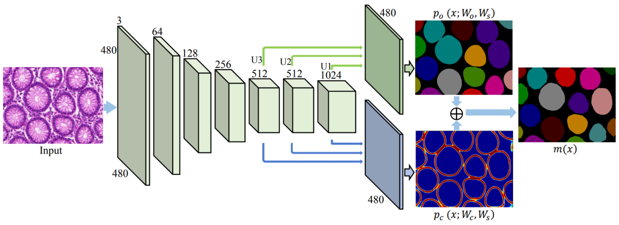

Fig 2. DCAN model architecture

2.2 Evaluation Metrics

2.2.1 Dice Coefficient (DICE):

Measures the overlap between predicted and ground truth masks. A higher Dice score indicates better segmentation performance.

$$Dice = \frac{2|X \cap Y|}{|X| + |Y|}$$

2.2.2 Intersection over Union (IoU):

Calculates the ratio of the intersection to the union of predicted and ground truth regions. Commonly used to evaluate segmentation accuracy.

$$IoU = \frac{|X \cap Y|}{|X \cup Y|}$$

2.2.3 Hausdorff Distance (HD) & 95th Percentile HD (HD95):

HD measures the maximum boundary distance between prediction and ground truth. HD95 reduces sensitivity to outliers by focusing on the 95th percentile.

$$H(X,Y) = max\big(h(X,Y), h(Y,X)\big)\ where\ h(X,Y) = \underset{x \in X}{max}(\underset{y \in Y}{min}\|x - y\|), h(Y,X) = \underset{y \in Y}{max}(\underset{x \in X}{min}\|y - x\|)$$

2.2.4 Average Symmetric Surface Distance (ASSD):

Computes the average distance between the surfaces of predicted and true masks, in both directions. Lower values indicate more accurate boundary alignment.

$$ASSD(X,Y) = \frac{1}{|X| + |Y|}\left(\sum_{x \in X} \underset{b \in B}{min}\, h(x,y) + \sum_{y \in Y} \min_{x \in X} h(y,x)\right)$$

2.3 Preprocessing

2.3.1 Data Preparation

After downloading the dataset from Kaggle, we first organized the files into a clear and structured directory format.

2.3.2 Resize

In the base model, the original images, masks, and outputs are resized to 256×256. However, we believe this approach may distort the original shape of the objects, potentially affecting the model's prediction accuracy. To address this issue, we implemented pad-resize, which will be discussed in a later section.

2.3.3 Color normalization: Macenko

To reduce color inconsistency caused by different labs and equipment, we apply Macenko stain normalization. When enabled, a target image is loaded and standardized, and the normalizer is fitted to it. This ensures all images have a consistent color style for better model performance.

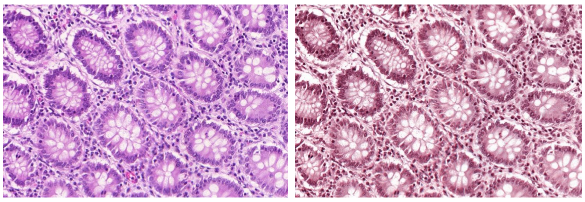

Fig 3. Original image (left) vs. Macenko-normalized image (right)

2.3.4 Data Augmentation

If the aug argument isn’t assigned, like in the base model, the input images are only normalized and then converted to PyTorch tensors. However, when aug is enabled, the images will undergo a series of data augmentation operations defined in self.transform.
These include a horizontal flip with a 50% probability, random adjustments to brightness and contrast with a 30% probability, and elastic transformation with a 20% probability, which simulates slight non-rigid deformations.

3 Experiments

3.1 Base Model

To verify the effectiveness of our chosen model architecture, we first experimented with a basic setup. The base model was trained on grayscale images that were directly resized to 256×256 pixels and trained for 30 epochs.

3.2 Different models

We experiment with two models: U-Net and DCAN. While U-Net is a strong baseline, DCAN is designed for better boundary prediction with its contour-aware dual-decoder structure. We expect DCAN to perform better, especially when objects are clustered or have unclear edges.

3.3 Different number of epochs

In the base model, we trained for only 30 epochs. To evaluate the impact of training duration, we compare the performance of models trained for 30 and 300 epochs. We expect that longer training can lead to better results by allowing the model to learn more refined features.

3.4 Grayscale vs RGB images input

In the original training process, we convert input images to grayscale to simplify the data and speed up training. However, we also explore using RGB images to investigate whether additional color channels can improve segmentation performance.

3.5 Different resize methods: direct resize vs pad-resize

As mentioned in the preprocessing section, we resize all images to 256×256. However, direct resizing may distort the original shape of objects in the image. To address this, we also experiment with pad-resize, which preserves the original aspect ratio by resizing the image proportionally and padding the remaining space to reach 256×256.

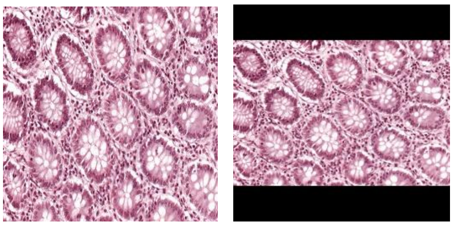

Fig 4. Direct resize (left) vs. Resize with padding (right)

3.6 Impact of TTA(Test-Time Augmentation)

TTA applies multiple augmentations to each test image, generates predictions for all augmented versions, and averages the results to produce a more reliable final output. We experiment with TTA to evaluate whether it can improve segmentation performance by reducing prediction variance and enhancing boundary accuracy.

4 Results & Analysis

4.1 U-Net & DCAN base

Tabel 1. Performance Comparison of U-Net and DCAN Across Evaluation Metrics

model	Dice	IoU	HD(pixel)	HD95(pixel)	ASSD(pixel)
U-Net base	**0.8439**	**0.7398**	58.6638	17.5872	2.7776
DCAN base	0.8407	0.7392	**52.3220**	**15.3345**	**2.5505**

Using U-Net with 30 epochs achieves a Dice score of 0.8439, indicating better overlap accuracy. In contrast, DCAN with the same number of epochs achieves a HD score of 52.3220, demonstrating its strength in capturing more precise boundary information.

4.2 U-Net

4.2.1 All methods

Tabel 2. Performance Comparison of U-Net under different model setting

model	Dice	IoU	HD (pixel)	HD95 (pixel)	ASSD (pixel)
base	0.8439	0.7398	58.6638	17.5872	2.7776
ep300	0.8556	0.7599	53.3327	10.5369	1.7509
base + resize	0.824	0.7175	44.3854	14.9751	2.5614
ep300 + resize	0.8458	0.7449	43.8974	14.163	2.2574
rgb ep30	0.8186	0.711	45.3115	12.1988	2.2212
rgb ep300	0.8675	0.7782	44.4096	7.5542	1.2965
rgb resize ep30	0.8685	0.7808	37.6719	10.3578	1.6777
rgb resize aug ep30	0.8742	0.7838	43.0691	10.2371	1.4563
rgb resize aug ep300	0.8844	0.8014	39.0133	6.2782	1.086

Table 2 presents the results of all U-Net configurations. The best performance is achieved using RGB input with resizing, data augmentation, and extended training, reaching a Dice score of 0.8844 and the highest boundary accuracy with an ASSD score of 1.086. The second and third best results also use RGB input, both achieving around 0.87 in Dice score.

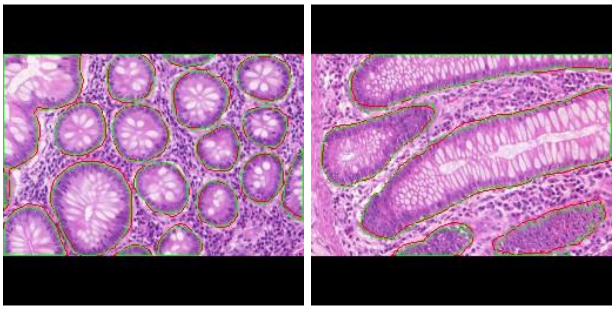

Fig 5.Visualization of segmentation boundaries produced by “RGB resize aug ep300” U-Net model (Red line: GT mask boundary / Green line: predicted mask boundary)

4.2.2 Epoch 30 vs Epoch 300

We observe that training for 300 epochs generally outperforms 30 epochs in both Dice and IoU scores. It also consistently reduces boundary distance metrics. This suggests that longer training enhances model performance. Moreover, since the training loss was still decreasing at 300 epochs, it is likely that even better results could be achieved with further training.

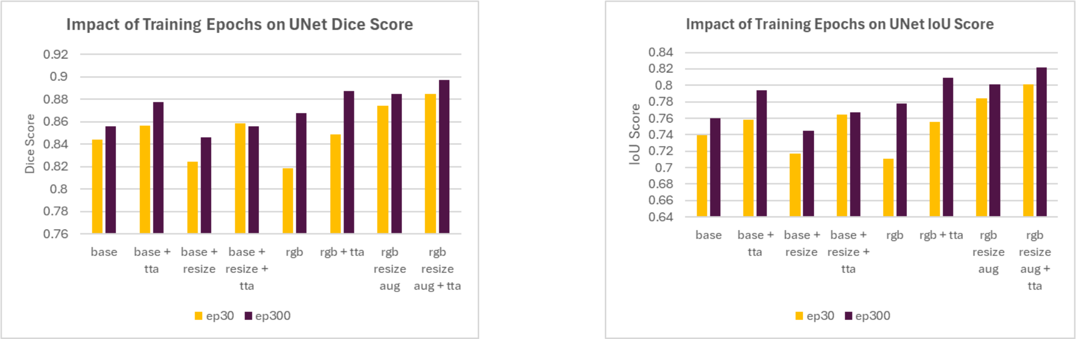

Fig 6 & 7. Impact of training epochs on U-Net Dice and IoU score across different settings

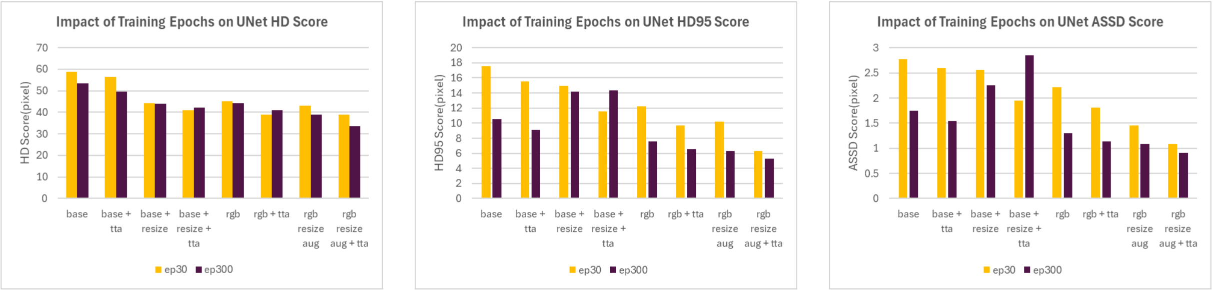

Fig 8 ~ 10. Impact of training epochs on U-Net HD, HD95 and ASSD across different settings

4.2.3 Gray vs RGB

At the base level, grayscale input slightly outperforms RGB. However, when training for 300 epochs and using resize and TTA, RGB data improves the model’s generalization ability. We also see that RGB models consistently achieve lower boundary scores, meaning smoother segmentations.

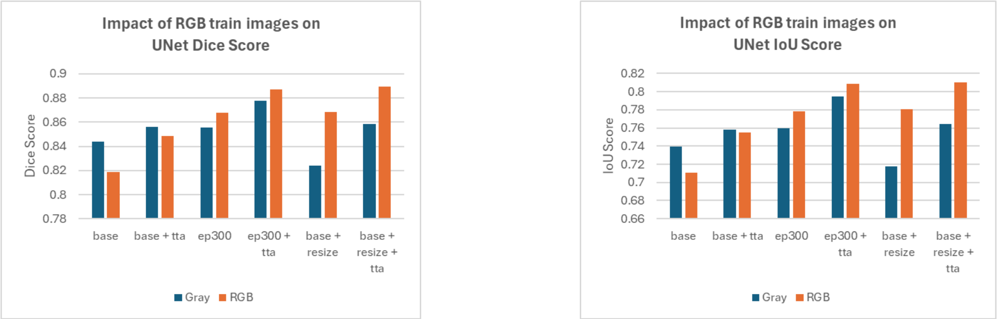

Fig 11 & 12. Comparison of Grayscale and RGB Input on U-Net Dice and IoU Scores

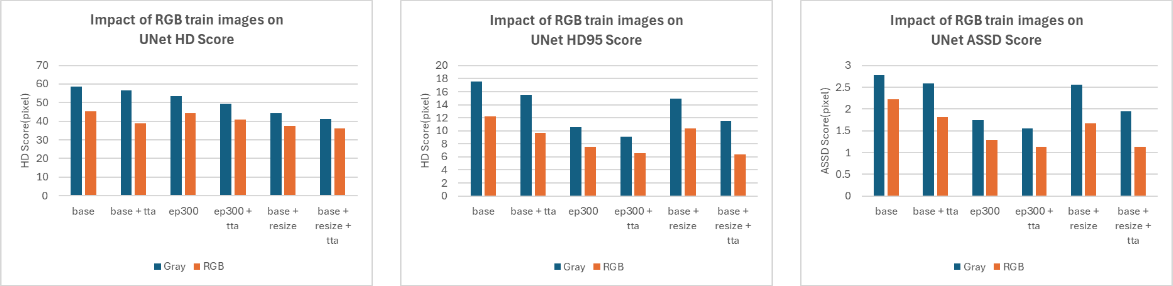

Fig 13 ~ 14. Impact of Grayscale vs. RGB on U-Net HD, HD95 and ASSD

4.2.4 Direct resize vs. Pad-resize

We see that for most cases in grayscale, direct resize actually performs better on area metrics, except for base + TTA. In boundary accuracy, pad-resize performs better for HD but worse for ASSD in the 300-epoch models.

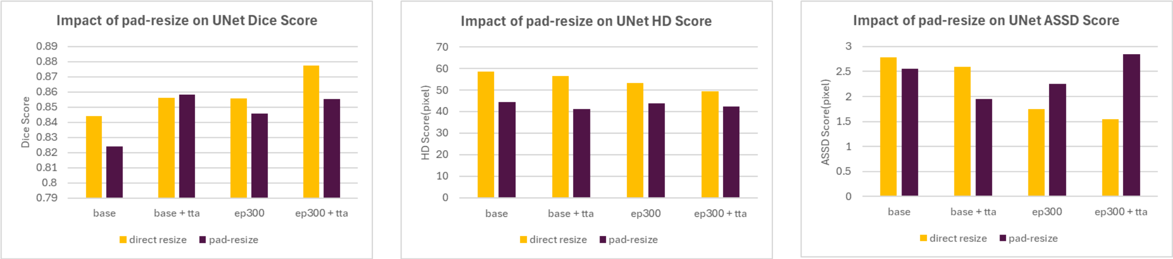

Fig 15 ~ 18. Performance comparison of resize methods on U-Net Dice, HD and ASSD

Additionally, when comparing grayscale and RGB images, pad-resize boosts RGB performance because color gradients help distinguish padding from real edges. Grayscale only has intensity, so the padding confuses the model.

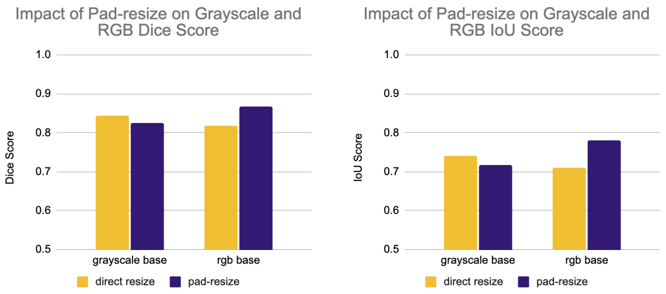

Fig 19 & 20. Effect of Pad-resize on Grayscale and RGB images in U-Net Dice and IoU Score

4.2.5 With vs Without TTA

It is obvious to see that TTA improves Dice and IoU scores by around 2-4%, and also reduces boundary scores.

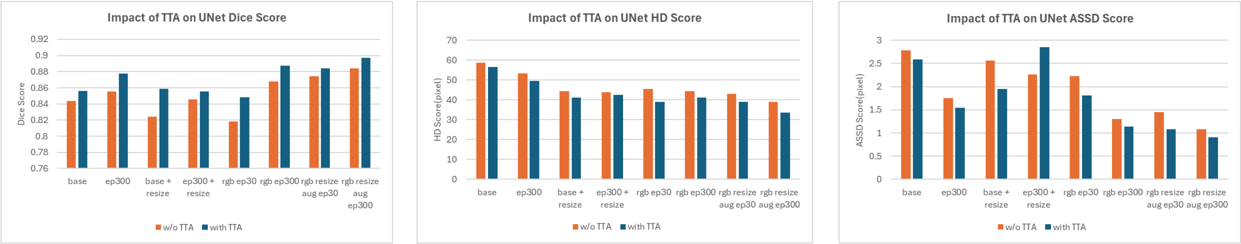

Fig 21 ~ 23. Effect of TTA on U-Net Performance: Dice, HD, and ASSD Metrics

In table 3, The best results come from the RGB model with resize augmentation and TTA, achieving the highest Dice score 0.8969 and lowest HD 33.5581. Overall, TTA helps produce more accurate and robust segmentation results.

Tabel 3. Performance Comparison of U-Net model with and without TTA

model	Dice	IoU	HD (pixel)	HD95 (pixel)	ASSD (pixel)
rgb ep300	0.8675	0.7782	44.4096	7.5542	1.2965
rgb resize aug ep30	0.8742	0.7838	43.0691	10.2371	1.4563
rgb resize aug ep300	0.8844	0.8014	39.0133	6.2782	1.086
rgb ep300 + tta	0.8872	0.8088	40.9608	6.5984	1.1374
rgb resize aug ep30 + tta	0.8844	0.8014	39.0133	6.2782	1.086
rgb resize aug ep300 + tta	**0.8969**	**0.8214**	**33.5581**	**5.276**	**0.9053**

4.3 DCAN

4.2.1 All methods

In table 4, we can see that the best area metric is achieved by the rgb 300 epoch method of Dice score 0.8831. While best boundary predictions are from a different model that add resize and augmentation, achieving 38.2794 HD score.

Tabel 4. Performance Comparison of DCAN under different model setting

model	Dice	IoU	HD (pixel)	HD95 (pixel)	ASSD (pixel)
base	0.8407	0.7392	52.3220	15.3345	2.5505
ep300	0.8722	0.7843	51.073	9.8453	1.8192
rgb ep30	0.8743	0.7868	41.8043	13.7098	2.0924
rgb ep300	**0.8831**	**0.7989**	49.6056	12.3183	1.7844
rgb resize aug ep30	0.8786	0.7917	37.3987	11.6201	1.597
rgb resize aug ep300	0.8772	0.7912	**38.2794**	**6.3488**	**1.0812**

These visual results in Fig 25. show that the DCAN's dual-decoder successfully captures both object localization and boundaries.

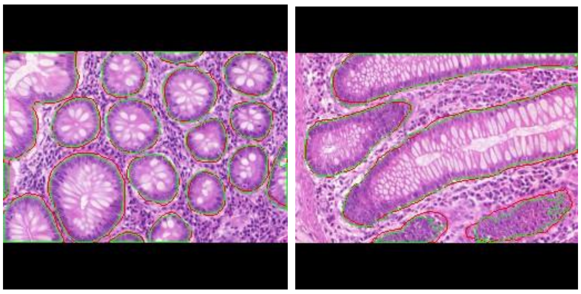

Fig 24. Visualization of segmentation boundaries produced by “RGB resize aug ep300” DCAN model (Red line: GT mask boundary / Green line: predicted mask boundary)

4.2.2 Gray vs RGB

We can see that using RGB gives a 3% boost for the base model and around 1% boost for others in Dice score. For boundary metrics, RGB data clearly helps improve the results compared to grayscale.

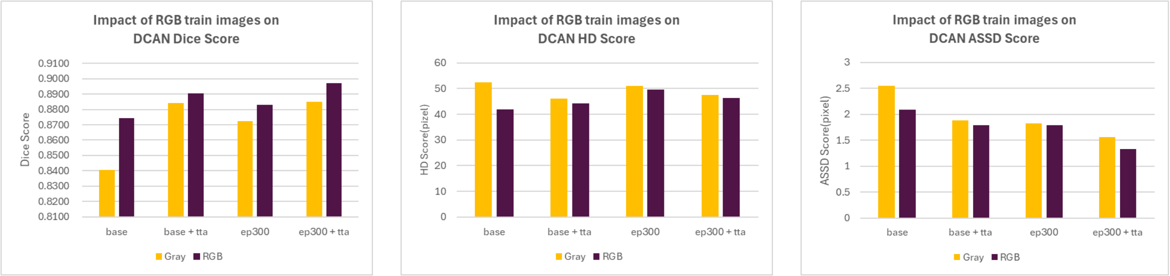

Fig 25 ~ 27. Impact of Grayscale vs. RGB on DCAN Dice, HD and ASSD

4.2.3 With vs Without TTA

It can be observed that TTA brings about a 1% Dice score improvement overall. IoU also benefits, with increases from 1 to 3%. TTA also helps with boundary performance. This shows that TTA effectively reduces noise and improves model performance.

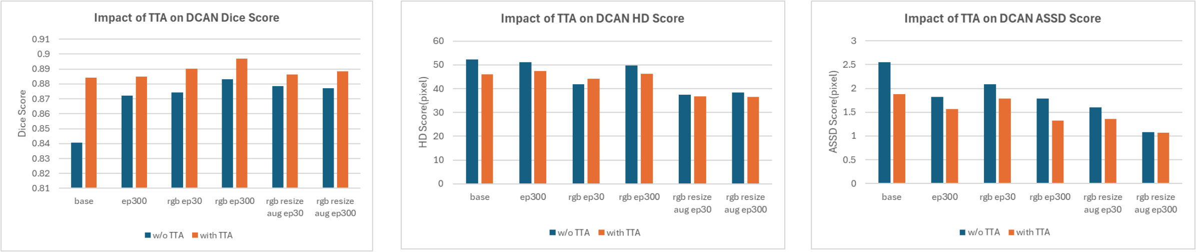

Fig 28 ~ 30. Impact of TTA on DCAN Dice, HD and ASSD

We can see that applying TTA to DCAN leads to the highest Dice score of 0.8971. TTA also helps improve boundary predictions. Combined with other techniques, we can reach a HD of 36.6038 and an ASSD of 1.065.

Tabel 5. Performance Comparison of DCAN model with and without TTA

model	Dice	IoU	HD (pixel)	HD95 (pixel)	ASSD (pixel)
rgb ep300	0.8831	0.7989	49.6056	12.3183	1.7844
rgb resize aug ep30	0.8786	0.7917	37.3987	11.6201	1.597

rgb resize aug ep300	0.8772	0.7912	38.2794	6.3488	1.0812
rgb ep300 + tta	**0.8971**	**0.8206**	46.3536	8.8439	1.3252
rgb resize aug ep30 + tta	0.8864	0.8036	36.7794	10.2739	1.355
rgb resize aug ep300 + tta	0.8885	0.8092	**36.6038**	**5.947**	**1.065**

4.4 Different models (U-Net vs. DCAN)

For area metrics like Dice, DCAN generally performs better, except in the resize group. As for HD95, DCAN is expected to perform better, and it does on grayscale images, but when using RGB input, UNet actually outperforms DCAN. We suspect this is because additional color may distract DCAN from focusing on precise boundaries.

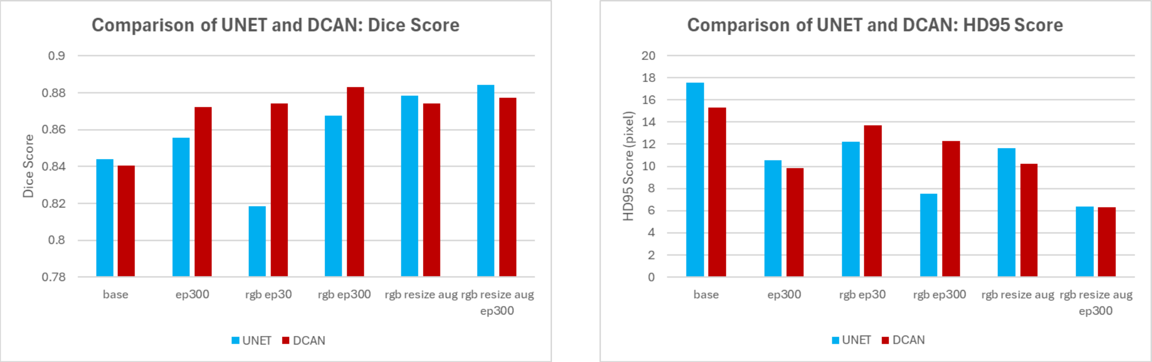

Fig 31 & 32. Performance Comparison between U-Net and DCAN model

U-Net generally produces better overlap performance, as shown in Fig. 34 (left). However, DCAN demonstrates superior boundary precision, particularly when two cells are close together. In such cases, DCAN is able to separate them, whereas U-Net may mistakenly predict them as a single connected region, as illustrated in Fig. 35 (right).

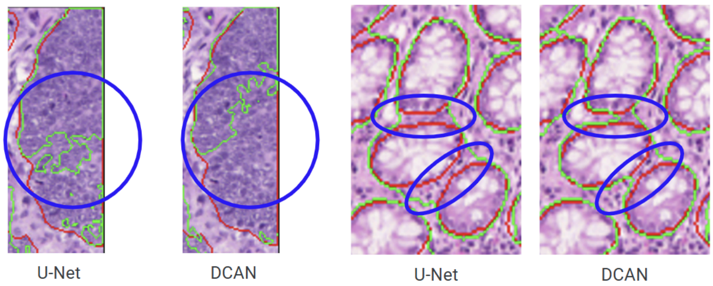

Fig 33 & 34. Prediction results for U-Net and DCAN

5 Conclusion & Future Works

We summarize our experimental results as follows:

- Data Augmentation
  - Pad-resize hurts Grayscale model, but enhances RGB model.
  - TTA improves performance in most model settings.
- Model Settings
  - With data augmentation and longer training, RGB models outperforms grayscale models.
  - Models trained for 300 epochs generally outperforms 30 epochs.
- Model Structure
  - In early training stages, DCAN has better boundary predictions compared to UNet.

    - Eventually, UNet can catch up to DCAN, especially with RGB images and data augmentation.
- Best performance
    - [DCAN rgb ep300 + tta] Dice: 0.8971
    - [UNET rgb resize aug ep300 + tta] HD: 33.5581

In the future, we plan to improve our models further by:

- Train the models for more epochs to exhaust the network's capacity
- Add training / validation set to avoid over fit
- Try hybrid architectures (eg. U-Net++) for richer feature extraction
- Try more boundary loss functions
- Try different tuning strategies for the boundary loss of DCAN
- Experiment with more kinds of data augmentation techniques (eg. Cutmix)
- Post-processing with Morphological Operations (e.g., erosion, dilation) using scipy.ndimage helps clean noise and fill gaps

6 Learnings

Through this project, we learned how to preprocess medical images, including techniques such as color normalization. We also became familiar with evaluating segmentation results using both area-based and boundary-based metrics. In addition, we gained hands-on experience in implementing and training U-Net and DCAN models. We also explored different loss functions and learned how to combine multiple losses using weighted sums to better balance area and boundary accuracy.

7 References

- Dataset: https://www.kaggle.com/datasets/sani84/glasmiccai2015-gland-segmentation/data
- Gland Segmentation in Colon Histology Images: The GlaS Challenge Contest: https://arxiv.org/pdf/1603.00275v2
- Gland segmentation task with GlaS 2015 dataset using UNet model: https://github.com/twpkevin06222/Gland-Segmentation
- U-Net: Convolutional Networks for Biomedical Image Segmentation: https://arxiv.org/abs/1505.04597
- DCAN: Deep Contour-Aware Networks for Accurate Gland Segmentation: https://arxiv.org/abs/1604.02677