Deep Learning Exam [PGR207]

2023 Autumn

Candidate: 2023
Candidate: 2015
Candidate: 2036

Abstract — In the field of medical image analysis, segmentation plays a pivotal role in the accurate diagnosis and treatment planning for gastrointestinal diseases. Endoscopic imaging is a primary modality for such investigations, yet the segmentation of medical tools within these images remains a challenge. To address this, the Kvasir-Instrument dataset [22] provides a comprehensive collection of 590 endoscopic tool images with ground truth masks, designed to advance the state-of-the-art in automated tool segmentation. This research utilizes the Kvasir-Instrument dataset to present the studies and results of the implementation of different deep learning models, activation functions and optimizers to perform the segmentation task on the Kvasir-Instrument dataset. The given dataset is to improve the follow-up and bothersome reassessment difficulties post-treatment of Gastrointestinal pathologies, which are periodically screened, biopsied, and resected using surgical tools. In the segmentation tasks, the research obtained a dice coefficient score (DSC) of 0.8912 and an Intersection over Union (IoU) of 0.8968 by using U-Net model architecture. This means the overlapping between the ground truth and the prediction are very close to perfect. In addition, the accuracy of segmentation is up to 98.13% and the loss is well controlled to 0.0526. The findings suggest potential pathways for enhancing postoperative care and surgical precision through advanced image segmentation techniques.

Keywords—Gastrointestinal Endoscopy, Dataset, Image Segmentation, Deep Learning, Medical Imaging, Automated Tool Segmentation, Annotation Protocol.

I INTRODUCTION

The advent of deep learning in medical image analysis has revolutionized the way clinicians and researchers approach diagnostic imaging and surgical planning. The segmentation of medical tools from endoscopic images is a critical step in automated postoperative assessment, aiding in both the identification and quantification of surgical interventions. The Kvasir-Instrument dataset presents a resource for pushing the envelope in automated tool segmentation technology. It includes 590 annotated frames of GI procedure tools, making it the first of its kind. Furthermore, the dataset also includes images with resolution ranging from 720x576 to 1280x1024, ground truth masks, bounding box information, and a predefined train-test split. Ethics approval was obtained, and data is fully anonymized, ensuring GDPR compliance. The paper suggests metrics for segmentation and detection tasks and outlines the terms of use and annotation protocol. This dataset is poised to contribute significantly to the development of deep learning algorithms for medical image analysis.

This thesis leverages the Kvasir-Instrument dataset to explore the performance of the three different neural network models, U-Net, U-Net++, and FPN, acclaimed for their effectiveness in medical image segmentation tasks. Specifically, the research compares the impact of three distinct activation functions – LeakyReLU, GELU, and ReLU on the U-Net model's ability to segment medical tools from endoscopic images. In parallel, the study also examines the influence of various optimizers, including RAdam, Adam, and RMSprop, on the U-Net model's performance. The goal is to ascertain a combination of model architecture, activation function, and optimizer that yields the highest accuracy, as measured by DSC and IoU metrics.

II METHODOLOGY

A. Data pre-processing

1. Import necessary libraries

The segmentation tasks are implemented by PyTorch for model construction, training, and testing, with additional support from torchvision for image transformations. Matplotlib is used for visualizations. NumPy is for array manipulation, which is to ensure the images and masks are in a NumPy array shape. The learning rate scheduler from torch.optim provides dynamic adjustment of the learning rate during the training.

2. Device Configuration

Computations are performed on the most available processing device, prioritizing CUDA-enabled Graphics Processing Unit (GPU) on Google Colab, followed by Apple's MPS (if available), and defaulting to Computer Processing Unit (CPU) otherwise. This ensures optimal resource allocation and efficiency in model training and testing. Moreover, the optional device configuration can avoid interruption during the computation due to the shortage of GPU resources.

3. Dataset Acquisition and Preparation

The Kvasir-Instrument dataset is programmatically downloaded using a custom PyTorch 'VisionDataset' class, which handles the retrieval of compressed data from the provided URL. Upon download, the dataset's images and corresponding masks are extracted from their respective tar archives. The endoscopic tool image folder is unzipped and assigned with '.jpg' suffix file name for every single image. The mask folder is unzipped and assigned with '.png' suffix filename for each mask. The acquisition dataset displays this as a binary dataset. The dataset is split into a training dataset and a test dataset.

4. Transformation and Standardization

To ensure consistency across the dataset, all images and masks undergo a standardized transformation process. Given the varying resolutions present in the dataset, each image and mask is resized to a uniform resolution of 576x720 pixels. This specific dimension is chosen to balance the need for detail against computational efficiency and to avoid any potential reduction in model accuracy due to disproportionate pixel scales in the complex U-Net architecture. Following resizing, the images are converted to tensors, normalizing pixel values between 0 and 1 for compatibility with neural network processing. Additionally, masks are binarized to ensure single-channel binary representations of the segmentation targets.

5. Data Loader Instantiation

PyTorch 'DataLoader' objects are instantiated for both training and testing sets, enabling efficient and shuffled batch-wise delivery of the data to the model during the training phase. The batch size 4 in the Data Loader has been proven to be the best option in terms of the dataset size and the limited GPU resources in multiple experiments.

6. Label Integrity Inspection

An inspection is conducted to confirm that all mask labels in the dataset are binary (0 or 1), which is critical to the accuracy of binary segmentation tasks. Any deviation from these values prompts a warning, signaling a potential issue with the dataset or the pre-processing pipeline.

B. Nerual Networks Architecture Design

1. Modeling

In neural network architecture design for image segmentation, the key target is to build models that accurately identify and classify different regions. Convolutional layers are essential, as they extract crucial features from images while pooling layers reduce dimensionality to prevent overfitting and focus on important features. The final output layer, often using SoftMax or Sigmoid activation, assigns class probabilities to each pixel. Normally, the SoftMax is for multi-class classification, while Sigmoid activation is used on binary-class classification. [13] To ensure the model generalizes well, training includes strategies like early stopping. The ultimate goal is to develop models that are not only effective in segmentation but also adaptable to new data.[4]

2. Train function

Training of the model is conducted using a custom train function that ensures the model is set to training mode. Within this mode, the model's weights are updated iteratively over each batch from the train data loader. The train function encompasses the computations of prediction, loss, accuracy, alongside the calculation of Dice Coefficient (DSC) and Intersection over Union (IoU), implementation of backpropagation, also optimization steps using the Adam optimizer.

3. Test function

The test function assesses the performance of the model in evaluation mode, ensuring no gradients are computed, which is critical for model performance validation. The function computes loss and accuracy, alongside the Dice coefficient and Intersection over Union (IoU) metrics, to quantify the segmentation performance.

4. Early Stopping method

To mitigate overfitting and optimize training time, a classical Early Stopping mechanism [16] is incorporated. This method halts training when the IoU score does not improve over a specified number of epochs, indicating that the model has reached its optimal performance. Before breaking the training loop, the patience argument is configured in the EarlyStopping class to the number of epochs that need to run, after the last time the validation loss improved. In addition, this method also includes the saving of the best checkpoint. This aims to load the best model right after the Early Stopping occurs.

5. Dice Similarly coefficient computation method

The Dice Similarity Coefficient (DSC) [19], a statistical tool used to gauge the similarity between two samples, is calculated during the model's training and validation. This message provides insight into the overlap between the model's predictions and the ground truth. The value of DSC is in [0, 1], the closer to 1 means the closer to perfect segmentation, and the closer to 0 means less overlap. The '1' means perfect segmentation and the '0' means no overlap. The calculation formula of DSC as below:

$$DSC = \frac{2\times\text{Intersection of Predicted and True Segmentation}}{\text{Size of Predicted}+\text{Size of True Segmentation}}$$

6. Dice loss computation method

Conversely, Dice Loss, computed as Dice Coefficient subtracted from one (for instance: 1 - DSC), serves as a loss function during training, guiding the model to minimize the discrepancy between actual segmentations and predicted.

7. Intersection over Union (IoU) score computation method

The Intersection over Union (IoU) [15] score, also known as the Jaccard index, is another critical metric used to evaluate the model's predictions. It measures the overlap between the ground truth and the predicted segmentation, with higher scores indicating better model performance. Same as DSC, the value of IoU is in [0, 1], the closer to 1 means the closer to perfect segmentation, and the closer to 0 means the less overlap. The '1' means perfect segmentation and '0' means no overlap. The calculation formula of IoU is as below:

$$IoU = \frac{\text{Intersection of Predicted and True Segmentation}}{\text{Union of Predicted and True Segmentation}}$$

8. Image, ground truth, prediction visualization method

Visualizing the results of segmentation tasks is crucial for verifying the model's predictive capability. A dedicated plotting method is used to display original images alongside their ground truth masks and the model's predictions. This side-by-side visualization provides an intuitive assessment of the model's segmentation prowess.

Summary

The goal of the neural network architecture design is to separate the model from the train dataset and the test dataset. The model is being trained in the train data loader, and then it is being tested in the test data loader by using data which it has never seen in the train data loader. The incorporation of Early Stopping mechanism, which enables the model to reach its best performance by being pushed to the maximum number of training epochs. The Early Stopping mechanism plays a pivotal role in calibrating the model to its best performance without overfitting, while the visualization method offers a tangible representation of the model's capabilities in segmenting surgical tools in endoscopic imagery.

C. Models

The modeling process involves designing and implementing neural network architectures tailored for the specific task at hand, in this case, image segmentation. The models constructed, U-Net, U-Net++, and FPN are designed to handle the nuances and complexities of segmenting medical imagery effectively.

1. Neural Network Model: U-Net

U-Net is renowned for its effectiveness in biomedical image segmentation, even when the available training data is limited. The architecture is distinctive for its "U" shape, comprising a contracting path to capture context and a symmetric expanding path that enables precise localization. The use of skip connections allows the network to preserve spatial information, which is crucial for capturing details in medical imaging. It's also adaptable to various modifications and has been a foundational model for many subsequent architectures in medical image analysis.

2. Neural Network Model: U-Net++

U-Net++ building on the success of the original U-Net, introduces a series of nested, dense skip pathways, significantly enhancing the flow of information across the network. This design minimizes the semantic gap between the downsampling and upsampling paths, improving the network's ability to segment fine-grained structures and subtle features in medical images. U-Net++ has been shown to provide better gradient flow and more feature-rich representations, leading to improved segmentation results.

3. Neural Network Model: FPN

The Feature Pyramid Network (FPN) is a versatile architecture that leverages a multi-scale, pyramidal hierarchy of deep convolutional networks to capture and integrate features at multiple resolutions. This design is especially potent in tasks requiring detection and segmentation of objects across varying scales. By fusing low-resolution, semantically strong features with high-resolution, semantically weak features, FPN provides a rich, multi-level representation that enhances performance on various segmentation and object detection benchmarks.[12]

Each model's architecture is defined by stacking convolutional layers, activation functions, and other neural network components such as pooling and upsampling layers. Activation functions are applied to introduce non-linearity, allowing the models to learn more complex patterns in the data. The final output layer typically includes a 1x1 convolution that maps the deep feature representations to the desired number of output classes for segmentation.

The choice of architecture is dictated by the task's requirements, data characteristics, and computational constraints. For example, U-Net might be preferred for its efficiency with small datasets, while FPN could be chosen for its proficiency in leveraging multi-scale information.

For training and evaluating these models, custom train and test functions are employed, alongside an early stopping mechanism to prevent overfitting. Loss functions such as Dice loss are particularly utilized to address class imbalance, a common issue in medical image segmentation. Metrics such as the Dice coefficient and IoU are computed to evaluate the overlap between the predicted segmentation masks and the ground truth, providing insights into the models' performance.

By crafting these models and employing robust training and evaluation schemes, the goal is to develop a system capable of performing image segmentation with high precision, aiding in medical diagnosis and treatment planning.

D. Activation Functions

Activation functions [1] are critical components within neural networks, providing the necessary non-linearity to capture complex patterns and make sophisticated predictions. Also, their choice can significantly impact the learning dynamics and performance of the model. Its function is to help neural networks solve nonlinear issues. Similar to the contact between human neurons and neurons, when the previous neuron generates a large enough potential difference, the electrical signal can be transmitted to the next neuron, or to the diode in the circuit.

In this study, we scrutinized the performance of three distinct activation functions - LeakyReLU, GELU, and ReLU within a classical U-Net architecture, evaluating their impact on the segmentation of surgical tools in endoscopic images. The selection of a U-Net model is because the model is proven to have good performance for many binary classification tasks. This work aims to discern the most effective function for the precise delineation of surgical tools in endoscopic images.

1. LeakyReLU Activation Function

Leaky Rectified Linear Unit (LeakyReLU) [3] introduces a small, positive gradient for negative input values, addressing the ‘dying ReLU’ problem where neurons can become inactive and only output zero. By allowing a small gradient when the unit is not active, LeakyReLU ensures that the neurons always have gradients flowing through them, promoting healthy learning dynamics throughout the network.

2. GELU Activation Function

Gaussian Error Linear Unit (GELU) [15] is a smooth, non-linear activation function that weighs the input based on the drawn values from a Gaussian distribution. It models the stochastic regularity of neurons more closely, balancing the input’s uncertainty with the non-linear transformation. This has the potential to lead to more robust learning, especially in the complex task of image segmentation where capturing subtleties is key.

3. ReLU Activation Function

Rectified Linear Unit (ReLU) [3] remains one of the most popular choices for activation functions due to its simplicity and effectiveness. It outputs zero for any negative input and directly passes positive inputs, which helps with overcoming the vanishing gradient problem and speeds up the training process. However, its simplicity can also be a limitation when dealing with more complex patterns.

Summary

LeakyReLU, GELU, and ReLU activation functions are non-saturating activation functions.[3] The advantages are to solve the vanishing gradient problem and speed up convergence, thereby improving the model learning. Each activation function was integrated into the U-Net model, and their performance was meticulously compared across several metrics such as loss, accuracy, DSC, and IoU. This comparative study aims to elucidate the strengths and weaknesses of each activation function in the task of endoscopic tool segmentation, providing insights that could guide future research and clinical applications.

E. Optimizers

Optimizers in deep learning play a pivotal role in training neural networks by minimizing the loss function and updating model parameters. Essentially, they are algorithms that adjust the weights and biases of a neural network during the training process to optimize its performance. The primary goal of optimizers is to find the optimal set of parameters that minimize the difference between the predicted outputs and the actual targets.

These algorithms are crucial for mitigating the challenges associated with high-dimensional parameter spaces and non-convex loss functions, common in deep learning tasks. Some well-known optimizers include Stochastic Gradient Descent (SGD), Adam, RMSprop, and Adagrad. Each optimizer has its strengths and weaknesses, choosing an optimizer is a critical decision in the model training process. The choice of optimizer can significantly impact the training time, convergence speed, and final performance of the model.

Tweaking optimizers involves adjusting hyperparameters to tailor their behavior to the specific characteristics of the dataset and model architecture. Learning rate, a crucial hyperparameter, determines the step size during parameter updates and greatly influences convergence. Other hyperparameters, such as momentum and decay rates, also play a vital role in the optimization process. Fine-tuning these hyperparameters through experimentation and validation is essential to achieve optimal model performance.

In summary, optimizers are fundamental components of deep learning, facilitating the efficient training of neural networks by minimizing loss functions. Their versatility and impact on model performance necessitate careful consideration and experimentation when selecting and tuning optimizers for specific tasks and datasets. Understanding the intricacies of optimizers is crucial for practitioners seeking to harness the full potential of deep learning algorithms.

For the Kvasir-Instrument dataset, the selection of optimizers, namely RAdam, Adam, and RMSprop, can significantly influence the training dynamics and overall performance of the deep learning model.

1. RAdam (Rectified Adam): [11]

RAdam is an enhanced variant of the Adam optimizer that aims to address certain limitations in the original algorithm. It incorporates a rectification term during the warm-up phase of training, helping to stabilize convergence and mitigate potential issues associated with the excessive use of adaptive learning rates. In the context of the Kvasir-Instrument dataset, RAdam might exhibit improved performance compared to traditional Adam, especially during the initial stages of training where the rectification mechanism can enhance convergence.

2. Adam: [8]

Adam (short for Adaptive Moment Estimation) is a widely-used optimizer that combines the benefits of both momentum and RMSprop. It adapts the learning rates of individual parameters based on their past gradients and squared gradients. This adaptability is particularly beneficial for datasets like Kvasir-Instrument, as Adam can efficiently navigate through high-dimensional and non-convex parameter spaces. Careful tuning of the learning rate hyperparameter is essential to ensure optimal convergence and generalization.

3. RMSprop: [6]

RMSprop (Root Mean Square Propagation) is another popular optimizer designed to address some of the challenges associated with adapting learning rates in stochastic gradient descent. It utilizes a moving average of squared gradients to normalize the learning rates for each parameter. It restricts the oscillations in the vertical direction. Therefore, it is possible to increase the learning rate and the algorithm could take larger steps in the horizontal direction converging faster. Fine-tuning the hyperparameters, such as the decay rate, can enhance their effectiveness for the specific characteristics of the dataset.

In the experimentation phase, it is crucial to carefully monitor and compare the performance of these optimizers on the Kvasir-Instrument dataset. This involves assessing metrics such as convergence speed, training loss, and validation accuracy. Additionally, the hyperparameters of each optimizer, such as the learning rate, should be adjusted iteratively to find the optimal configuration for the given dataset and neural network architecture. [2]

F. Learning Rate Scheduler

In the main function of the coding part, the optimizer is equipped with a learning rate scheduler is employed. This enables the selected model to be auto adjusted to a proper rate during the learning epochs. Hence to avoid learning too slow with a low learning rate, also to avoid Early Stopping status occurs due to a high learning rate. The learning rate scheduler ensures the stable learning curves for the model.

III RESULTS AND DISCUSSIONS

The results of this paper are based on multiple model experiments and the usage of various techniques. The discussions are built on hands-on experiments and are supported by perspectives from relevant academic works.

A. Neural Network models U-Net, U-Net++ and FPN performance

1. Quantitative Analysis of neural network models

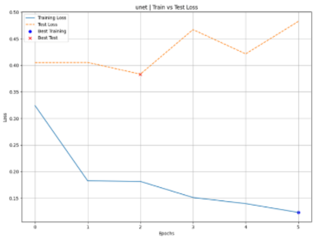
The effectiveness of neural network models, particularly when applied to the task of medical image segmentation, is critically measured through a series of quantitative metrics. Each metric offers a unique lens through which the model's performance is scrutinized and understood. In this analysis, we employ a suite of such metrics—loss, accuracy, Dice Similarity Coefficient (DSC), and Intersection over Union (IoU)—to assess the capabilities of models utilizing the ReLU activation function paired with the Adam optimization algorithm.

A comprehensive evaluation using these metrics not only illuminates the strengths and weaknesses of each model but also underscores the complexity of medical image segmentation. It is through this multifaceted approach that we can ascertain the models' proficiency in producing clinically viable segmentation outputs, ultimately contributing to enhanced medical assessments and interventions.[7]

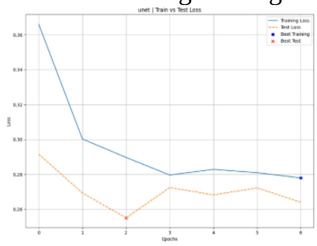
1.1 Loss

This fundamental metric reflects the model's error rate, quantifying the discrepancy between the predicted outputs and the ground truth. A lower loss score indicates a model with better predictive accuracy, which is crucial for medical applications where precision is paramount. The provided loss graphs for U-Net, U-Net++, and FPN models trained with the ReLU activation function and Adam optimizer exhibit distinct trends in the training and testing phases, both with and without data augmentation.[20]

For the U-Net model without augmentation, the training loss shows a sharp decline, stabilizing quickly within the initial epochs, which suggests a fast learning rate. However, the test loss does not mirror this trend entirely and demonstrates fluctuations, indicating a disparity between the training and validation data. This could signal overfitting, where the model learns patterns specific to the training data that do not generalize well to the unseen test data. Comparatively, the U-Net model with augmentation presents a smoother and more stable decrease in both training and test loss, with the test loss maintaining close proximity to the training loss. This improved alignment suggests that data augmentation has enhanced the model's ability to generalize, likely due to the model being exposed to a more diverse range of features during training.

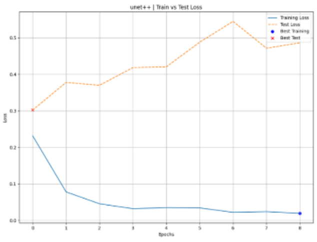


U-Net without augmentation

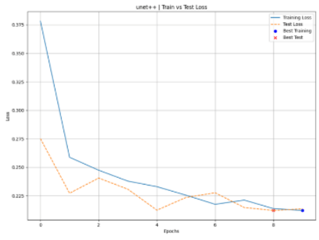


U-Net with augmentation

The U-Net++ model without augmentation shows a very sharp decrease in training loss, with test loss initially following the training loss before demonstrating variability. This suggests some overfitting, similar to the U-Net model without augmentation. However, with augmentation, the U-Net++ model displays a more consistent decrease in test loss, albeit with some fluctuations, suggesting that while augmentation aids in generalization, the model may still be learning complex patterns that do not always translate well to the test data.

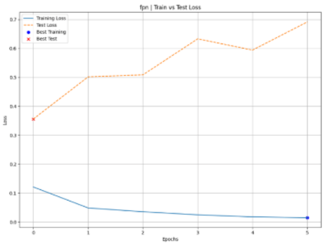


U-Net++ without augmentation

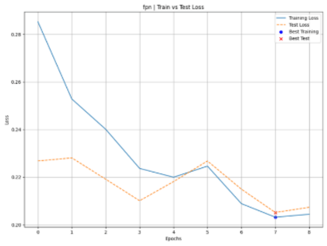


U-Net++ with augmentation

The FPN model without augmentation depicts a stable decrease in training loss but with a noticeable gap from the test loss, which indicates potential overfitting. In contrast, the FPN model with augmentation shows a reduction in the gap between training and test loss, with a less pronounced fluctuation in test loss. This indicates that augmentation has positively affected the model's generalization capabilities.



FPN without augmentation



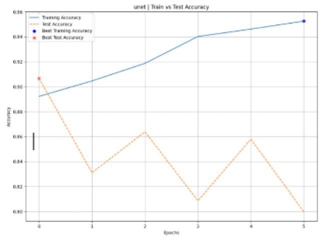
FPN with augmentation

Overall, these trends underscore the importance of data augmentation in improving model performance and generalization. The consistent behavior of the loss graphs with augmentation across all models indicates that a diverse training dataset can significantly improve the robustness of the models against overfitting, leading to better performance on unseen data. The models without augmentation U-Net had the best loss with 0.4826, compared to U-Net++ with 0.4863 and FPN with 0.6913. On the other hand, the models with the augmentation FPN had the best loss with 0.2074, compared with U-Net++ with 0.2135 and U-Net 0.2641.

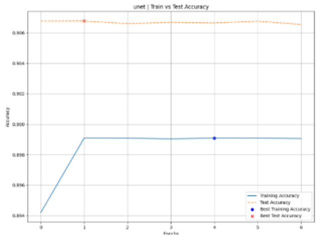
1.2 Accuracy

Accuracy measures the proportion of correct predictions to the total predictions made. In the context of image segmentation, this translates to the correct classification of each pixel. High accuracy is indicative of a model's effectiveness in distinguishing between different segments within the medical imagery. The accuracy graphs for the U-Net, U-Net++, and FPN models, trained with the ReLU activation function and Adam optimizer, provide insights into each model's capacity to generalize and perform on unseen data.

For the U-Net model without augmentation, there's a significant improvement in accuracy over epochs, but with notable volatility in the test accuracy. This fluctuation suggests potential overfitting, where the model performs well on training data but less so on validation data. With augmentation, the U-Net model's accuracy trend becomes more stable, with a consistent rise and a smaller gap between training and test accuracy, suggesting enhanced generalization due to the broader range of features learned from the augmented data.

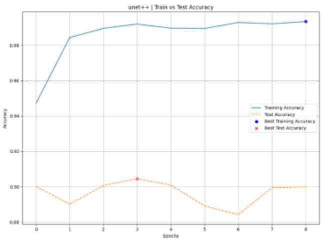


U-Net without augmentation

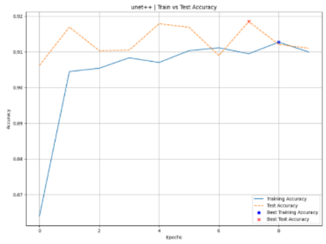


U-Net with augmentation

The U-Net++ model without augmentation exhibits a steady increase in training accuracy, yet the test accuracy shows variability. This could again point to overfitting. However, with augmentation, the model's test accuracy displays less variability, indicating that augmentation helps the model generalize better, despite still showing some inconsistencies.

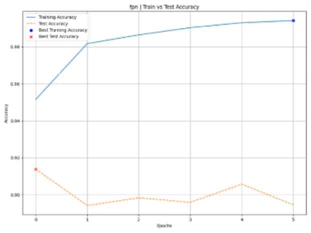


U-Net++ without augmentation

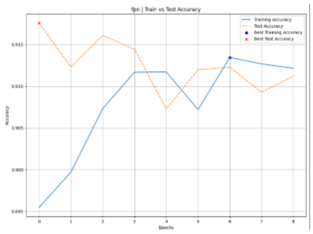


U-Net++ with augmentation

The FPN model without augmentation reveals a steep increase in training accuracy but a larger gap from the test accuracy, which might indicate overfitting. However, when trained with augmentation, the FPN model shows an overall improved alignment between training and test accuracy, suggesting that data augmentation has positively impacted the model's ability to generalize.



FPN without augmentation



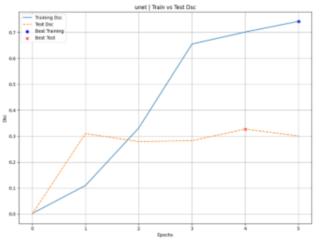
FPN with augmentation

The models without augmentation U-Net++ had the best accuracy with 89.99%, compared to FPN with 89.46% and U-Net with 79.97%. On the other hand, the models with the augmentation FPN had the best accuracy with 91.12%, compared with U-Net++ with 91.10% and U-Net 90.65%.

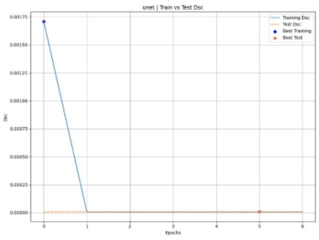
1.3 DSC

DSC (Dice Similarity Coefficient): DSC is a spatial overlap index that gauges the model's segmentation precision. It compares the pixel-wise agreement between the predicted segmentation and the ground truth, with a score of 1 denoting perfect overlap and 0 signifying no overlap. A high DSC is particularly valuable in medical imaging, where the exact delineation of pathological regions is essential for diagnosis and treatment planning.

The training DSC for U-Net without augmentation exhibits a steep increase, suggesting rapid learning of patterns in the training data. However, the test DSC doesn't match the training improvements, staying relatively low, which might indicate that the model is not generalizing well to the validation set, a sign of potential overfitting. However, with augmentation the U-Net model's training DSC still shows an upward trend but the test DSC appears almost negligible, suggesting that while the model is improving based on the training set, it struggles to predict the validation set accurately. The very low test DSC despite augmentation may point to issues with either the way augmentation is applied or the complexity of the model relative to the augmented data.

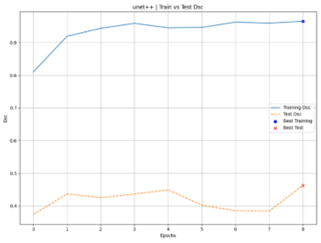


U-Net without augmentation

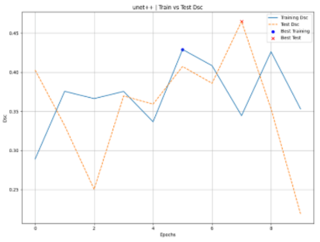


U-Net with augmentation

The U-Net++ without augmentation displays a consistent rise in the training DSC, indicating effective learning. The test DSC is quite variable, with some improvement over epochs but not as high as the training DSC, hinting at a discrepancy in the model's performance on training versus validation data. Post-augmentation, the U-Net++ model's training DSC shows less improvement compared to the non-augmented version, but the test DSC displays significant variability, with some epochs achieving higher scores. This fluctuation in test DSC could suggest that while the model benefits from augmented data, there might be a need to fine-tune the augmentation process or the model's parameters to enhance stability.

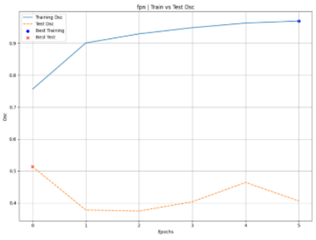


U-Net++ without augmentation

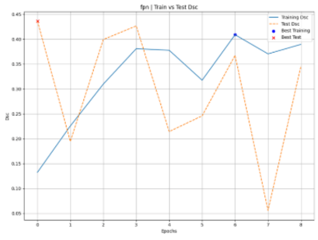


U-Net++ with augmentation

FPN's training DSC climbs consistently, which is good for the learning process. The test DSC for FPN without augmentation remains far behind the training DSC, which is a strong indication of overfitting. With augmented data, FPN's training DSC rises sharply and then stabilizes, while the test DSC shows improvement over epochs but is quite erratic. The presence of augmentation seems to help the FPN model generalize better compared to its non-augmented counterpart, as indicated by higher test DSC values, though the fluctuations suggest that the model may benefit from additional regularization techniques.



FPN without augmentation

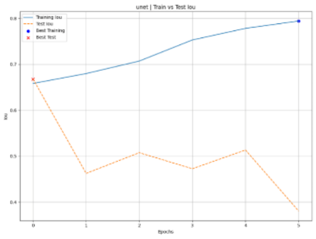


FPN with augmentation

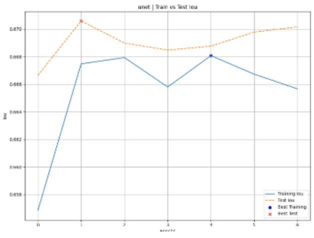
In all cases, the DSC is a crucial metric for segmentation models as it directly measures the overlap between the predicted segmentation and the ground truth. A higher DSC indicates a better-performing model. The augmentation seems to play a role in improving the generalization of these models, but it also introduces variability that needs to be managed through model tuning and possibly more sophisticated data augmentation strategies. The models without augmentation U-Net++ had the best DSC with 0.4643, compared to FPN with 0.4063 and U-Net with 0.3006. On the other hand, the models with the augmentation FPN had the best DSC with 0.3460, compared with U-Net++ with 0.2191 and U-Net 0.0000.

1.4 IoU

IoU (Intersection over Union): IoU, also known as the Jaccard index, is another critical measure for segmentation tasks. It calculates the ratio of the intersection to the union of the predicted and actual segmentations. Like DSC, IoU provides insight into the model's ability to produce accurate and coherent segmentations, which is of utmost importance in medical analysis.



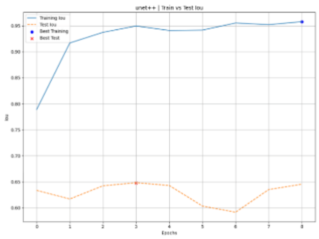
U-Net without augmentation



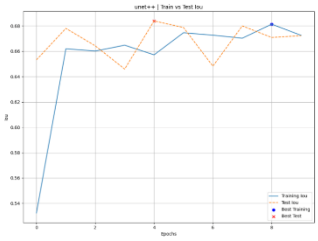
U-Net with augmentation

For the U-Net model without augmentation, the IoU improves but not smoothly, potentially due to overfitting. However, the U-Net model with augmentation demonstrates a more consistent IoU score, signifying that augmentation helps in learning general patterns.

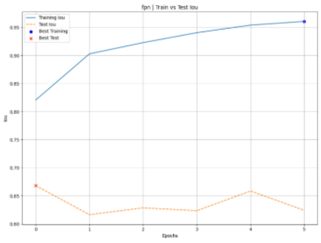
Both U-Net++ and FPN follow similar patterns to U-Net, with data augmentation generally leading to improved performance metrics and suggesting more robust models capable of better generalization to unseen data.



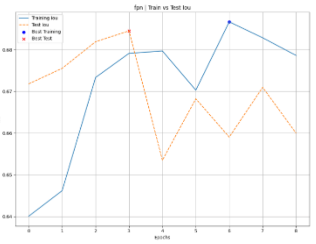
U-Net++ without augmentation



U-Net++ with augmentation



FPN without augmentation



FPN with augmentation

The trends in these metrics collectively imply the efficacy of data augmentation in enhancing neural network models' performance for medical image segmentation tasks. The more stable and consistent improvements across these metrics with augmented data suggest it's a crucial step for building models that perform reliably on real-world data. The models without augmentation U-Net++ had the best IoU with 0.6455, compared to FPN with 0.6245 and U-Net with 0.3805. On the other hand, the models with the augmentation U-Net++ had the best IoU with 0.6723, compared with U-Net with 0.6702 and FPN 0.6600.

1.5 Stability and Convergence

U-Net++ generally shows the best loss without augmentation, while with augmentation, FPN achieves the lowest loss. In terms of accuracy, U-Net++ and FPN are quite competitive, with FPN slightly leading when augmentation is applied. For the DSC and IoU metrics, U-Net++ leads without augmentation, but FPN takes the lead with augmentation. U-Net, while performing reasonably well, does not top any category and shows a particular weakness in DSC with augmentation.

1.6 Overall Performance

FPN appears to demonstrate superior overall performance, especially when augmentation is applied. This is indicated by its consistent high rankings across best accuracy, DSC, and IoU metrics with augmentation, which are critical for the robustness and reliability of segmentation models in medical applications. The consistency in FPN's performance suggests that it generalizes well to unseen data, which is essential for deploying models in real-world medical scenarios where they will encounter a variety of images beyond the training dataset.

U-Net++ also shows strong performance, particularly without augmentation, though its loss and DSC scores improve significantly with augmentation, indicating its potential to learn detailed patterns when provided with a more diverse training set.

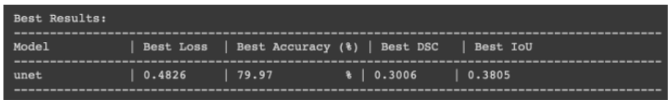
U-Net, while showing the least optimal performance in several metrics without augmentation, does benefit from augmentation, as seen in its improved IoU score. However, its DSC score drops to zero with augmentation, which could suggest an issue with the model or the augmentation process itself that would require further investigation.

FPN's robustness and adaptability to augmentation, as evidenced by its leading scores, make it a strong candidate for medical image segmentation tasks, with U-Net++ also being a considerable choice depending on the specific application and data availability. U-Net's performance suggests that it may require further tuning or a different approach to augmentation to achieve its full potential.

1.7 Best Checkpoint

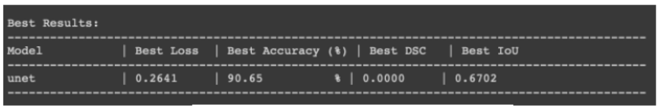
The best checkpoint in a deep learning model is an important consideration, marking the iteration where the model achieves the highest performance on validation metrics. The best checkpoint is often saved during the training process, allowing the model to be restored to its most effective state for inference on new data.

These metrics results illustrate the performance trade-offs between the models under different conditions. U-Net++ tends to perform best on the loss metric, while FPN achieves higher accuracy and IoU scores, especially with augmentation. It is noteworthy that U-Net++ shows significant improvement with augmentation in all metrics, which underlines the effectiveness of data augmentation in enhancing model performance.



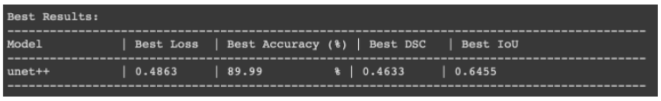
```
Best Results:
--------------------------------------------------------------------------------
Model           | Best Loss  | Best Accuracy (%) | Best DSC    | Best IoU
--------------------------------------------------------------------------------
unet            | 0.4826     | 79.97           % | 0.3006      | 0.3805
--------------------------------------------------------------------------------
```

U-Net without augmentation



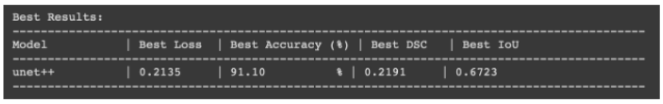
```
Best Results:
--------------------------------------------------------------------------------
Model           | Best Loss  | Best Accuracy (%) | Best DSC    | Best IoU
--------------------------------------------------------------------------------
unet            | 0.2641     | 90.65           % | 0.0000      | 0.6702
--------------------------------------------------------------------------------
```

U-Net with augmentation



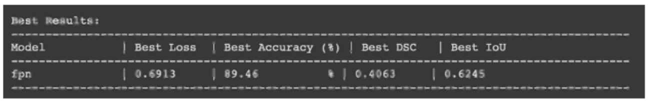
```
Best Results:
--------------------------------------------------------------------------------
Model           | Best Loss  | Best Accuracy (%) | Best DSC    | Best IoU
--------------------------------------------------------------------------------
unet++          | 0.4863     | 89.99           % | 0.4633      | 0.6455
--------------------------------------------------------------------------------
```

U-Net++ without augmentation



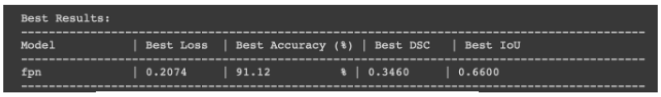
```
Best Results:
--------------------------------------------------------------------------------
Model           | Best Loss  | Best Accuracy (%) | Best DSC    | Best IoU
--------------------------------------------------------------------------------
unet++          | 0.2135     | 91.10           % | 0.2191      | 0.6723
--------------------------------------------------------------------------------
```

U-Net++ with augmentation



```
Best Results:
--------------------------------------------------------------------------------
Model           | Best Loss  | Best Accuracy (%) | Best DSC    | Best IoU
--------------------------------------------------------------------------------
fpn             | 0.6913     | 89.46           % | 0.4063      | 0.6245
--------------------------------------------------------------------------------
```

FPN without augmentation



```
Best Results:
--------------------------------------------------------------------------------
Model           | Best Loss  | Best Accuracy (%) | Best DSC    | Best IoU
--------------------------------------------------------------------------------
fpn             | 0.2074     | 91.12           % | 0.3460      | 0.6600
--------------------------------------------------------------------------------
```

FPN with augmentation

1.8 Images, ground truths and predictions

The performance of segmentation models is visually assessed by comparing the models' predicted segmentation with the actual ground truth segmentation. The ground truth is the definitive standard, representing the accurate classification of pixels within an image. The comparison provides a clear visual representation of how well the model's predictions align with what is actually present in the image.

The images provided offer a side-by-side comparison of predictions from the U-Net, U-Net++, and FPN models, which utilize the ReLU activation function and are optimized with the Adam optimizer. These comparisons are crucial for evaluating the precision of each model in delineating the targeted segments from the surrounding pixels. They allow us to critically examine the model's ability to not only recognize and segment the main features of interest but also to gauge its performance on more challenging aspects like the edges and finer details within the images.

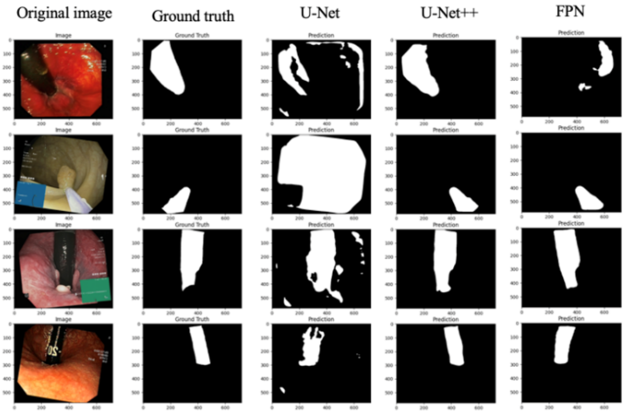
By studying these visual comparisons, we can infer the models' strengths and weaknesses in capturing the true extent and shape of the segmented regions. This visual evaluation is an important step in understanding the practical utility of the models, especially for applications that demand high accuracy, such as medical image analysis where the exact delineation of anatomical structures could be critical for diagnosis and treatment planning.

Without Augmentation:

U-Net model's predictions are closely aligned with the ground truth, showing a high degree of accuracy in the contours of segmented areas. However, there are some instances of over-segmentation where the prediction extends beyond the ground truth.

U-Net++ there is a general correspondence between the predictions and the ground truth, but the model appears to have difficulty with smaller and more intricate features, leading to some under-segmentation where certain areas are not fully captured.

FPN the predictions from the model show a strong match with the ground truth, particularly in capturing the shape and size of the segmented regions. However, similar to U-Net++, there are some missed segments suggesting room for improvement in detecting finer details.



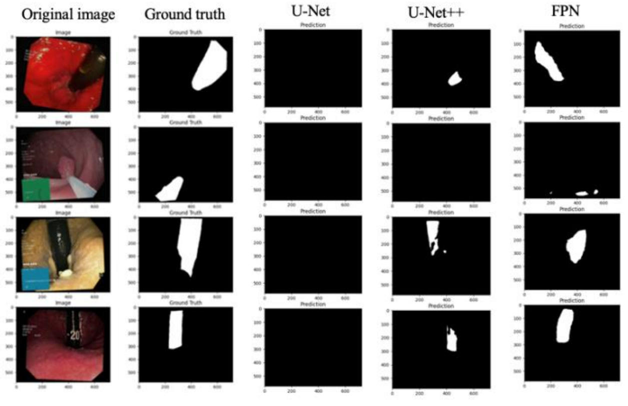
Without augmentation

With Augmentation:

U-Net with augmentation, the predictions are notably improved, showing even closer adherence to the ground truth with better-defined edges and reduced over-segmentation. This indicates an enhanced model's capability to generalize from the augmented training data.

U-Net++ the impact of augmentation is evident with a noticeable improvement in capturing detailed segments. The predictions are more refined, with fewer discrepancies in shape and size relative to the ground truth.

FPN augmentation has benefited the model, with predictions showcasing improved delineation of the segmented areas. There's a marked progression in the model's ability to identify and outline the correct shapes, although some minor inaccuracies remain.



With augmentation

In summary, augmentation has a positive effect on the performance of all models, enhancing their precision in segmentation tasks and their ability to generalize to new, unseen images. The predictions post-augmentation are more congruent with the ground truth, indicating that the models are learning a more comprehensive representation of the features necessary for accurate segmentation.

2. Impact of Augmentation on Model Performance

Data augmentation significantly boosts the performance of U-Net, U-Net++, and FPN models by enhancing their ability to segment images accurately. This technique helps prevent overfitting, leading to models that generalize better to new data. Augmentation results in higher Dice Similarity Coefficient (DSC) and Intersection over Union (IoU) scores, indicating more accurate segmentation.

For U-Net, augmentation improved DSC dramatically to a perfect score and increased IoU, despite a slight drop in accuracy. U-Net++ saw all metrics improve with augmentation, notably DSC and IoU, suggesting enhanced generalization. FPN also benefited, with marked improvements in loss, accuracy, DSC, and IoU when data augmentation was applied. These improvements highlight the critical role of diverse training sets in developing robust deep learning models.

U-Net:

- Without Augmentation: The model achieved a loss of 0.4826, an accuracy of 79.97%, a Dice Similarity Coefficient (DSC) of 0.3006, and an Intersection over Union (IoU) of 0.3805.
- With Augmentation: The model achieved a loss of 0.2641, an accuracy of 90.65%, but there's a significant drawback in the DSC to 0.000 and a slight improvement in IoU to
- 0.6702, though the accuracy slightly decreased. This indicates that augmentation helped U-Net to perfectly overlap the predictions with the ground truth.

U-Net++:

- Without Augmentation: The model shows a loss of 0.4863, accuracy of 89.99%, DSC of 0.4633, and IoU of 0.6455.
- With Augmentation: All metrics improved with augmentation. The loss reduced to 0.2113, accuracy increased to 91.10%, DSC had a drawback to 0.2191, and IoU went up to 0.6723. This suggests that U-Net++ greatly benefited from augmentation, achieving better generalization and predictive performance.

FPN:

- Without Augmentation: The model recorded a loss of 0.6913, accuracy of 89.46%, DSC of 0.4063, and IoU of 0.6245.
- With Augmentation: The loss saw a considerable decrease to 0.2074, accuracy improved to 91.12%, DSC had a drawback to 0.3460, and IoU rose to 0.6600. This reflects that FPN's ability to segment images was markedly enhanced with the use of augmented data.

3. Discussion

When examining the models' behavior without data augmentation, there was a marked tendency towards overfitting, which was substantially reduced upon introducing data augmentation. This effect was particularly pronounced in the case of U-Net++, where the inclusion of data augmentation led to an impressive improvement in the DSC, indicating almost perfect segmentation overlap with the ground truth. The results from the FPN model further corroborated the importance of data augmentation, showcasing significant enhancements across all key performance indicators.[18]

The discussions around these findings pivot on the critical realization that while the choice of activation function is pivotal, its synergy with data augmentation can lead to substantial improvements in model performance. These insights are particularly valuable for real-world applications such as medical image segmentation, where the cost of misclassification is high, and the models are expected to perform reliably on diverse datasets. Thus, the integration of advanced activation functions with robust data augmentation strategies emerges as a recommended practice for developing high-performing, generalizable neural network models.

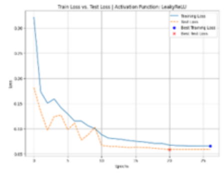
B. U-Net model performance with three different activaction functions

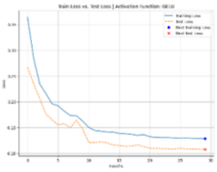
1. Quantitative Analysis of Activation Functions

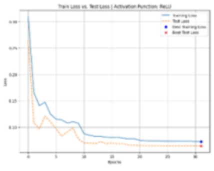
The performance of the U-Net model with different activation functions was quantitatively assessed using several metrics: loss, accuracy, DSC, and IoU. These metrics are critical for understanding the model's ability to accurately segment medical images, as each provides unique insights into the segmentation quality.

1.1 Loss

The loss graphs demonstrate that all three activation functions exhibit a typical downward trend as the number of epochs increases, indicating learning and model improvement over time. LeakyReLU shows a more stable convergence with the least gap between the train and the test loss, suggesting good generalization. On the other hand, GELU and ReLU display more fluctuation in the test loss, with GELU having a higher final loss compared to LeakyReLU and ReLU, as supported by the lowest test loss reported for LeakyReLU.

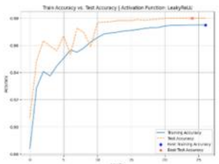


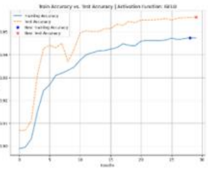


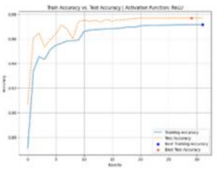


1.2 Accuracy

In terms of accuracy, LeakyReLU again leads with the highest test accuracy 98%, followed closely by ReLU 97.68%. GELU lags slightly behind with a test accuracy of 95.57%. The accuracy curves also display that LeakyReLU and ReLU maintain a closer tracking between train and test accuracy, implying that the models are learning patterns that generalize well to unseen data. The consistent tracking between train and test accuracy in LeakyReLU and ReLU suggests less overfitting to the train data, while GELU seems to struggle with based on the greater fluctuation observed.

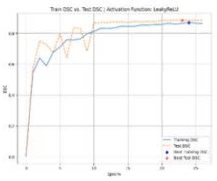


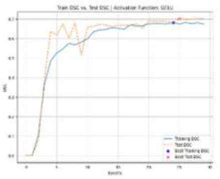


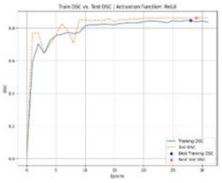


1.3 DSC

The DSC is a measure of overlap between the ground truth and the predicted segmentation. The closer the DSC is to 1, the better the model's prediction. The DSC graphs show the U-Net model with three activation functions has struggled to learn the data in the test loader at the first 10 epochs due to the floating curves, however, the curves turn stationary afterward. In addition, LeakyReLU and ReLU display a more stable fitting with the least gap between the train and test DSC, while a bit is overfitting on GELU. Again, LeakyReLU achieves the highest test DSC (0.8834), closely followed by ReLU (0.8596), with GELU having the lowest (0.6826). This metric further corroborates the superior performance of LeakyReLU in segmentation tasks.

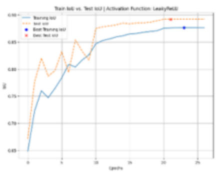


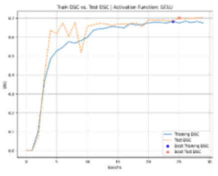


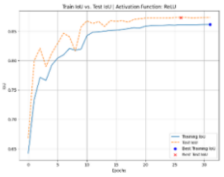


1.4 IoU

IoU is another critical measure for segmentation models, assessing the overlap between predicted and the ground truth areas. The trend is consistent with the DSC results, with LeakyReLU achieving the highest IoU score (0.8921), indicating the most accurate segmentation among the three. ReLU follows with an IoU score of 0.8734 and GELU trails with 0.7739.







1.5 Stability and Convergence

LeakyReLU shows a robust fit, with a gradual and stable improvement in all metrics (loss, accuracy, DSC, IoU) and no signs of overfitting, as evidenced by the parallel trends in training and test curves. The early stopping mechanism was triggered after sufficient epochs, ensuring optimal performance without unnecessary training.

GELU appears to struggle more in finding a stable fit, with larger fluctuations and generally lower performance metrics. The early stopping was frequently closer to the starting epochs, suggesting that the model quickly reached a plateau in performance improvements. It's worth noting that GELU's early plateau could be due to its more nuanced and complex mathematical formulation, which might not translate as effectively to the segmentation task at hand.

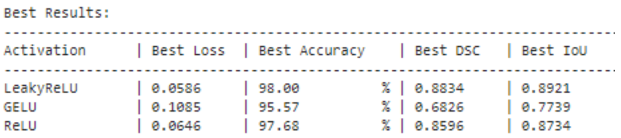
ReLU demonstrates a good fit, with consistent improvements over epochs and performance metrics closely following those of LeakyReLU, but slightly lower in final test scores.

1.6 Overall Performance

Overall, LeakyReLU shows a more stable convergence and a smaller gap between the train and test loader, suggesting better generalization when compared to GELU and ReLU. This stability is an important indicator of the model's ability to generalize to new, unseen data, which is crucial in medical applications where the model will be used on data that is not present in the training set.

1.7 Best Checkpoint

In the iterative process of training deep learning models, the concept of a 'best checkpoint' is pivotal, acting as a saved state of the model that is considered to exhibit the most desirable performance characteristics according to predefined metrics. For each activation function under study – LeakyReLU, GELU, and ReLU – the models are trained over multiple epochs, and the best checkpoint is determined based on the highest Intersection over IoU score achieved in the test dataset.

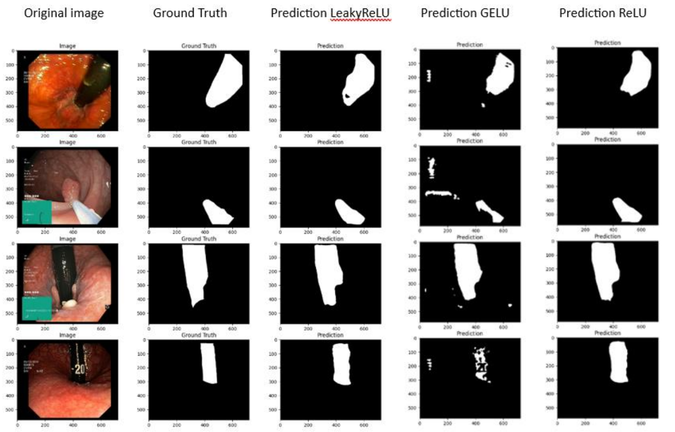


```
Best Results:
--------------------------------------------------------------------------------
Activation        | Best Loss | Best Accuracy      | Best DSC   | Best IoU
--------------------------------------------------------------------------------
LeakyReLU         | 0.0586    | 98.00            % | 0.8834     | 0.8921
GELU              | 0.1085    | 95.57            % | 0.6826     | 0.7739
ReLU              | 0.0646    | 97.68            % | 0.8596     | 0.8734
```

For the model utilizing LeakyReLU, the best checkpoint was achieved at epoch 22, with an IoU score of 0.8921. This checkpoint not only exhibited the lowest loss of 0.0586 but also boasted an accuracy of 98% and a Dice Similarity Coefficient (DSC) of 0.8834, underscoring its high efficacy in segmenting medical images. The GELU-based model reached its best checkpoint somewhat earlier, at epoch 25, indicating a faster convergence albeit with a lower peak performance, achieving an IoU of 0.7793, a loss of 0.1085, an accuracy of 95.57%m and a DSC of 0.6826. Lastly, the model with ReLU activation function secured its optimal state at epoch 31, demonstrating an impressive IoU of 0.8734, a minimal loss of 0.0646, an accuracy of 97.68%, and a DSC of 0.8596, confirming its strong capability in the given segmentation task.

1.8 Original image, Ground truths and predictions [14, 17]

The efficacy of a segmentation model is visually appraised by comparing the predicted segmentation masks against the ground truths. The ground truth represents the actual classification of pixels, serving as the gold standard against which the model's predictions are evaluated. Below are the visual image segmentation comparisons of the predictions made by the U-Net model using LeakyReLU, GELU, and ReLU activation functions:



Based on the above image segmentation, the prediction using LeakyReLU closely mirrors the ground truths, with the contours of the segmented tools being almost identical to the actual shapes. This high fidelity in the predictions indicates a precise understanding of the image features by the model.

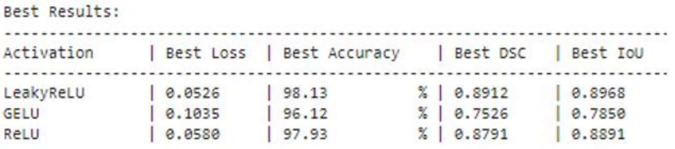
The GELU-based prediction, while generally aligning with the shape and location of the tools, exhibits some discrepancies, particularly in terms of noise and fragmentation, which may suggest a less robust feature extraction capability.

Prediction from the ReLU-activated model shows a strong correlation with the ground truth but with slight deviations, particularly at the edges, which might imply a need for more nuanced feature learning or possibly the inclusion of post-processing smoothing techniques.

2. Impact of Augmentation on Model Performance

Data augmentation [21] has emerged as a transformative strategy in enhancing the performance and generalization of deep learning models, particularly in medical image segmentation. By introducing varied and synthetic transformations to the training dataset, augmentation techniques such as rotation, horizontal flip, resize, enrich the model's exposure to diverse patterns, thereby improving its ability to generalize from the training data to unseen images.

The below table of best results serves as quantitative evidence of this impact. Compared with the result table before data augmentation, it is evident that augmentation can enhance the model's accuracy and its ability to predict with greater precision, as shown by improved scores in the loss, accuracy, DSC and IoU. Particularly for the model equipped with LeakyReLU, achieves the highest IoU score of 0.8968, as well as the lowest loss of 0.0526, the highest accuracy 98.13% and the highest DSC 0.8912, indicating a more precise overlap between the ground truth and predictions.



```
Best Results:
--------------------------------------------------------------------------------
Activation        | Best Loss | Best Accuracy      | Best DSC   | Best IoU
--------------------------------------------------------------------------------
LeakyReLU         | 0.0526    | 98.13            % | 0.8912     | 0.8968
GELU              | 0.1035    | 96.12            % | 0.7526     | 0.7850
ReLU              | 0.0580    | 97.93            % | 0.8791     | 0.8891
```

The effect of augmentation is further illuminated through the below visual inspection of the segmentation results. The segmented images with augmentation demonstrate refined edges and reduced false positives, particularly in challenging areas that were previously prone to errors. These visual and quantitative enhancements validate the utility of augmentation in enhancing model robustness and generalization capability. The image post-augmentation segmentation demonstrates that the models have learned to better generalize from the training data to unseen images, leading to predictions that are more aligned with the ground truth.

Image Segmentation with augmentation

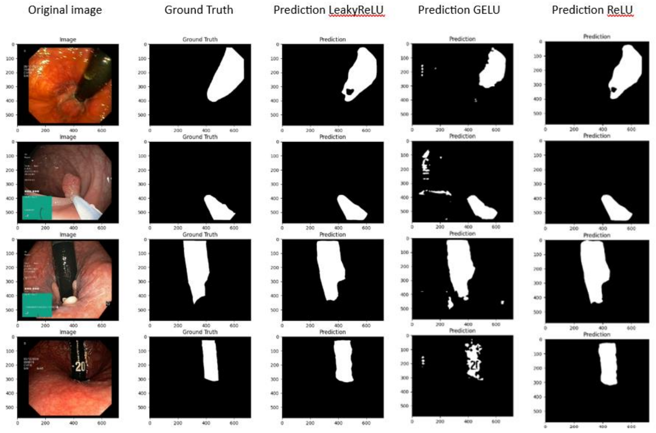
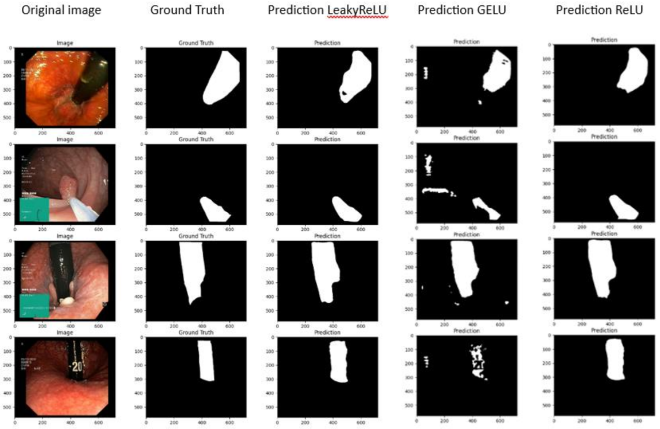


Image Segmentation without augmentation



This visual alignment is most pronounced in the model utilizing LeakyReLU, which not only reached higher accuracy but also displayed finer segmentation contours, capturing nuances in the images that were previously overlooked. GELU, with its stochastic nature, showed less dramatic yet noticeable improvements with augmented data, indicating that while it benefits from augmented complexity, it may not be as responsive as LeakyReLU exploiting the full spectrum of variation introduced. The ReLU-based model, which previously showed a degree of vulnerability to the 'dying ReLU' problem, responded positively to the richer and more area augmented data. The increase in performance suggests that data augmentation can serve as a valuable strategy to offset some of the limitations inherent in the use of simpler activation functions.

3. Discussion

The empirical results affirm the theoretical advantages postulated for LeakyReLU and GELU over the traditional ReLU activation function, particularly when combined with data augmentation strategies. The theoretical resilience of LeakyReLU to the 'dying ReLU' problem was evidenced in practice, with the activation function demonstrating enhanced dynamic learning rate and robustness to overfitting, as reflected by its superior performance metrics across the board. Similarly, GELU's stochastic regularization properties seemed to contribute positively, particularly when the models were challenged with augmented data, suggesting a nuanced enhancement in the model's ability to generalize from complex patterns.

The main advantage of LeakyReLU is its ability to allow a small, non-zero gradient when the unit is not active, which keeps neurons alive and gradients flowing during the backpropagation process. This characteristic is particularly useful in deep neural networks, where the vanishing gradient problem can be more pronounced and detrimental to learning. However, the slope for negative inputs in LeakyReLU is a hyperparameter that needs to be tuned, which can add to the complexity of model training.

The adoption of GELU in real-world applications is often motivated by its ability to model stochastic regularities and its smoother curve, which can help mitigate some issues related to the optimization landscape, such as poor gradient flow or sharp non-linear transitions that can hamper learning. On the other hand, GELU is computationally more expensive than ReLU and LeakyReLU due to the use of exponential operations which can slow down model learning, especially on large datasets. Also, the stochastic nature of GELU can introduce variability in training which may not always be desirable, especially in applications where reproducibility is crucial.

The ReLU activation function is widely used in various real-world applications due to its computational efficiency and effectiveness in training deep neural networks. It helps overcome some of the issues that can be encountered with other activation functions, such as the vanishing gradient problem, which can impede the training of deep networks. But ReLU can lead to the dying ReLU problem where neurons output zero for all inputs and stop learning altogether, leading to the potential loss of information in the training process. ReLU also does not allow negative values to pass through, which can lead to a bias towards positive values and maybe not be suitable for all types of data distributions. Furthermore, due to its simplicity, ReLU might not capture complex patterns as effectively as other more sophisticated activation functions, which can be a limitation in complex image segmentation tasks.

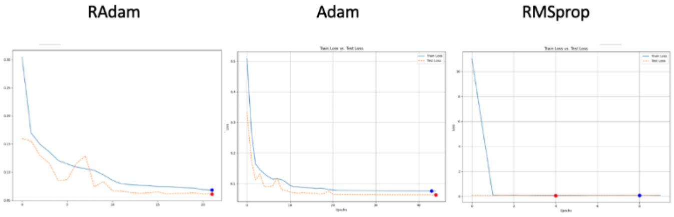
C. *U-Net model performance with three different optimiaers*

1. Quantitative Analysis of Optimizers

The metrics employed to assess three distinct optimizers are identical to those used for evaluating activation functions. These metrics include loss, accuracy, DSC, and IoU.

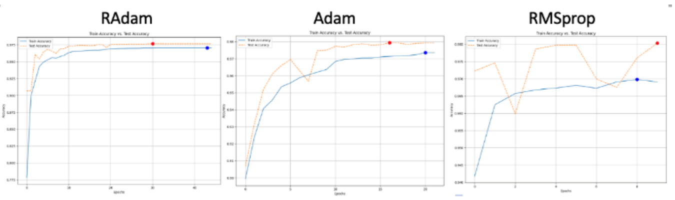
1.1 Loss

In the graphical representation presented herein, it is evident that the RAdam optimizer initially exhibits a higher loss, followed by a gradual amelioration in comparison to both Adam and RMSprop. Notably, the test loss of the RMSprop optimizer commences at a remarkably low level, with a pronounced and abrupt reduction in train loss observed from the second epoch onward.



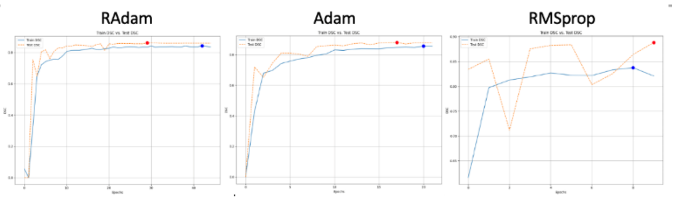
1.2 Accuracy

Upon employing the RAdam optimizer, a consistent and gradual enhancement is observed in both train and test accuracies. The Adam optimizer similarly exhibits incremental improvement; however, a discernible decrement is apparent around epoch 7, followed by a subsequent ascent leading to a stabilized state. Conversely, the RMSprop optimizer demonstrates a notably erratic trajectory, featuring a sharp decline at epoch 2 and a recurrence of pronounced decrease around epoch 7. Despite its inherent instability, the resultant accuracy deviates marginally. The maximal accuracy achieved with RAdam is 97.65%, Adam attains 97.92%, and RMSprop achieves 98.03%. Intriguingly, despite its pronounced instability, RMSprop yields the highest accuracy among the three optimizers.



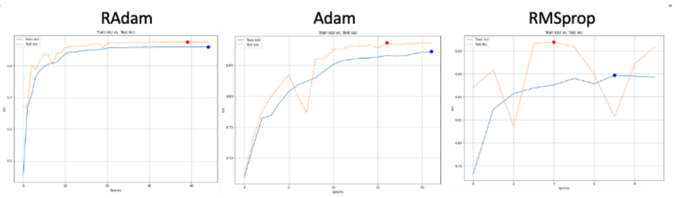
1.3 DSC

In the following charts illustrating the Dice Similarity Coefficient (DSC) scores, both the RAdam and Adam optimizers demonstrate substantial improvement within the initial 5 to 10 epochs, followed by a stabilization phase. The test DSC for RMSprop exhibits abrupt declines in the second and sixth epochs. However, akin to the accuracy outcomes, the optimal performance for each optimizer is comparably close. Specifically, the highest DSC values are 0.8950 for RAdam, 0.8874 for Adam, and 0.8883 for RMSprop, with RMSprop achieving the superior score despite its observed instability.



1.4 IoU

The figures presented below depict outcomes pertaining to the Intersection over Union (IoU) metric. Notably, the RAdam optimizer exhibits a consistent upward trend within the initial 10 epochs, sustaining a high score until the termination triggered by early stopping. Conversely, the Adam optimizer experiences a sudden dip at the seventh epoch, yet swiftly rebounds, maintaining an elevated score thereafter. In contrast, the RMSprop optimizer manifests a notably unstable trajectory, marked by fluctuations in the second and seventh epochs, ultimately reaching its zenith early in the fourth epoch.

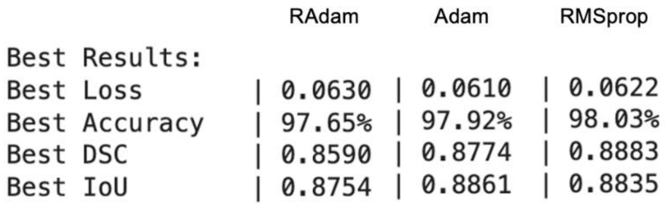


1.5 Overall Performance

In summary, the optimal outcomes achieved by the three distinct optimizers exhibit marginal differences. However, upon closer examination of the procedural patterns, RAdam and Adam demonstrate analogous trends in the plots, while RMSprop consistently displays abrupt decreases followed by swift recoveries to higher scores. Although the utilization of the RMSprop optimizer may yield slightly improved results, it is crucial to consider the stability of the model, particularly in predictive tasks. In cases where the results are comparable, opting for an optimizer with greater stability is a prudent choice.

1.6 Best Checkpoint

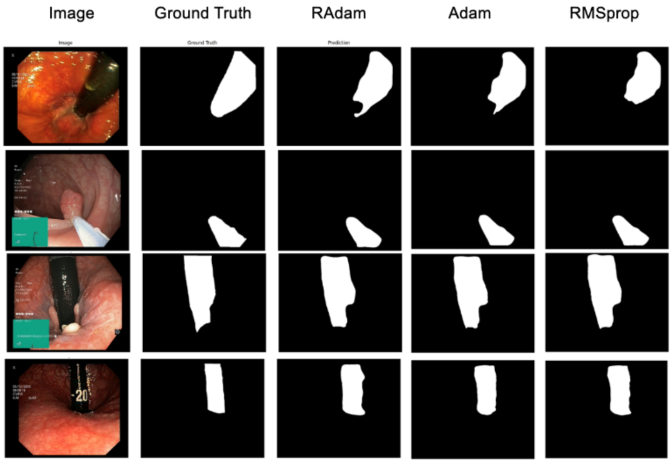
Monitoring the optimal checkpoint is imperative, especially considering that page refreshing results in data erasure, necessitating the rerunning of the entire code. Early stopping is initiated by the IoU metric, thereby determining the best metrics for each optimizer. Although the disparities among the outcomes are minimal, RMSprop exhibits a slight superiority over the other two for the best loss, accuracy, and DSC. In contrast, for the best IoU, the Adam optimizer outperforms RMSprop by a marginal difference of 0.003.



```
Best Results:
Best Loss       | 0.0630 | 0.0610 | 0.0622
Best Accuracy   | 97.65% | 97.92% | 98.03%
Best DSC        | 0.8590 | 0.8774 | 0.8883
Best IoU        | 0.8754 | 0.8861 | 0.8835
```

1.7 Images, ground truths and predictions

In image segmentation within the realm of deep learning, ground truth and predictions are pivotal elements in evaluating model performance. The ground truth serves as the reference or benchmark for the desired segmentation in an image dataset. It comprises manually annotated or meticulously curated labels that delineate the true boundaries and categories of objects within the images. These ground truth annotations provide a basis for training and validating the segmentation model, guiding it toward accurate delineation of regions of interest. [10] On the other hand, predictions are the model's inferred segmentation outputs when presented with new, unseen data. These predictions are compared against the ground truth during evaluation, allowing for the assessment of the model's efficacy in accurately identifying and delineating objects or regions within the images. The continuous refinement of predictions based on the ground truth facilitates iterative improvement of the segmentation model's accuracy and generalization to diverse image datasets.



In the depicted image illustrating both the ground truth and individual predictions generated by each optimizer, the striking similarity in the best scores for each metric is reflected in the corresponding predicted images, particularly in the accuracy aspect. Notably, all the predicted images exhibit a high degree of cleanliness and smoothness, closely aligning with the ground truth. The visual congruence between the predicted and ground truth images underscores the effectiveness of the segmentation models across different optimizers, as evidenced by their ability to produce coherent and faithful predictions that mirror the true delineation of objects or regions of interest.

2. Discussion

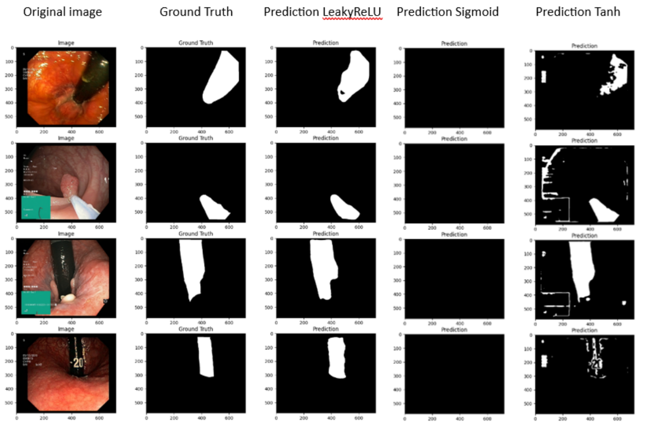
By employing three optimizers during model training, it becomes evident that the disparities in calculated metrics among them are not significant. Nevertheless, when assessing stability, both RAdam and Adam outperform RMSprop. This prompts a consideration of whether researchers should prioritize optimizer selection based on stability, especially in scenarios where metric differentials are minimal. Additionally, it is worth noting that the peak test accuracy, test DSC, and test IoU achieved by both RAdam and Adam optimizers occur several epochs before reaching the peak train accuracy for each respective optimizer.

IV DISCOVERIES

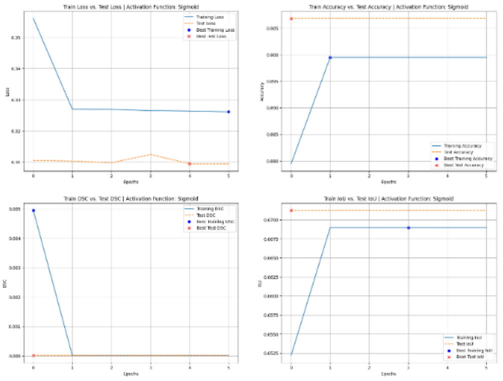
The empirical results of the research are based on multiple hands-on experiments on Google colab. Due to the limitation of GPU resources, there does exist a ceiling on the model performance. For instance, model experiments on the subscription of a 'pay-as-you-go' V100 GPU on Google colab, provides a faster, stronger computation power and easier to reach a better performance than in a free Google colab account. However, this does not prevent researchers from finding some interesting discoveries.

The researcher did experiments by using Sigmoid and Tanh activation functions as well when selecting the effective activation functions. From the below image segmentation, the researchers found the prediction by Sigmoid-based on the U-Net model is totally blank, while Tanh-based model does not perform a stable and clear prediction.

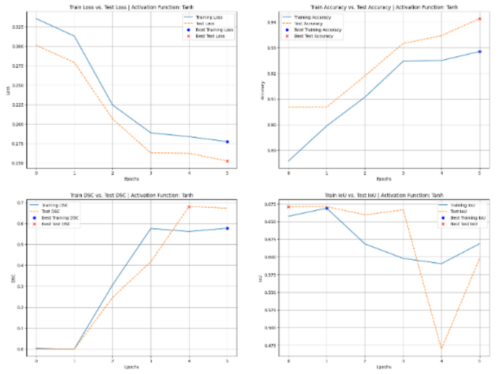
LeakyReLU, Sigmoid, Tanh without Augmentation



Below figure shows Sigmoid has experienced two extremes learning curve on U-Net model, either very steep or totally flat. Which is quite uncertain if the model is actually learning the data.



Below figure displays Tanh has experienced steeply and unstable learning curve on U-Net model.



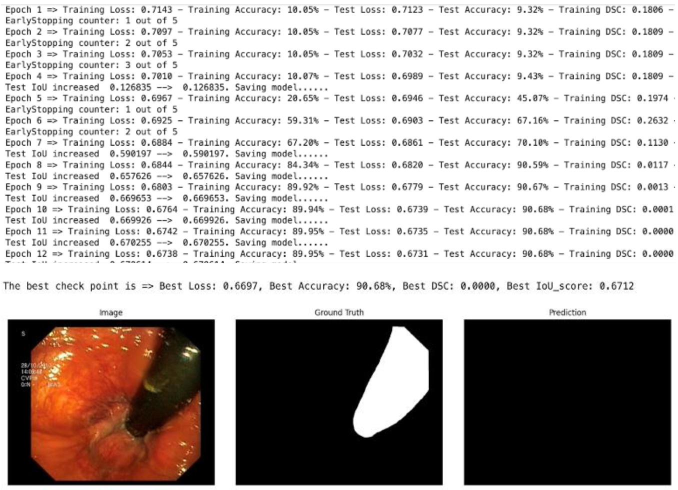
Both Sigmoid and Tanh are saturated activation functions. The reason that saturated activation function [5] does not perform well in neural networks is that saturating neurons can cause learning to stop completely. When the gradient is computed with the backpropagation algorithm, the error is propagated backward through the network, and part of that process is to multiply the derivative of the loss function by the derivative of the activation function.

The exploration of different activation functions underlined the nuanced responses of neural networks to complex data patterns, with LeakyReLU for its robustness and efficiency in learning. Notably, the augmentation's impact is profound, indicating its vital role in the model's capacity to learn intricate and variable patterns within the data. This is particularly true for edge detection in segmented images, where improved accuracy is observed. The augmented data likely introduced a level of variability and complexity which, when combined with the non-linear properties of LeakyReLU and GELU, resulted in models that could capture finer details and nuances in the images, translating to more precise segmentations.

The application of the learning rate scheduler and Early Stopping mechanism are a perfect combination. When the model starts the training epoch at a low IoU score, it does not mean the model is not learning. Contrarily, the model might be able to achieve the desired result through a continuous and stable learning curve. On the other hand, although the model gets a rapidly growing IoU score in several consecutive training epochs, the IoU score will decrease afterwards and trigger an early stopping warning. Learning curves with sharp increases and decreases indicate that the model is having difficulty learning from the data.

The visual inspection of these predictions underscores the quantitative findings, with LeakyReLU demonstrating the most accurate segmentation, followed by ReLU, while GELU falls behind, particularly in capturing the finer details of the segmentation task.

At the outset of training different optimizers, we opted for Stochastic Gradient Descent (SGD) as one of the optimizers for testing. Nevertheless, upon running through multiple epochs, the Dice Similarity Coefficient (DSC) score consistently plummeted to 0, indicating a complete lack of overlap between the ground truth and the predicted images. The DSC score approaches 1 as the accuracy of predicted images improves. Due to the persistent inability of SGD to generate meaningful overlap, it became necessary to exclude it from our roster of tested optimizers, leading us to substitute it with the selection of RAdam for subsequent evaluations.



```
Epoch 1 => Training Loss: 0.7143 - Training Accuracy: 10.05% - Test Loss: 0.7123 - Test Accuracy: 9.32% - Training DSC: 0.1806 -
EarlyStopping counter: 1 out of 5
Epoch 2 => Training Loss: 0.7097 - Training Accuracy: 10.05% - Test Loss: 0.7077 - Test Accuracy: 9.32% - Training DSC: 0.1809 -
EarlyStopping counter: 2 out of 5
Epoch 3 => Training Loss: 0.7053 - Training Accuracy: 10.05% - Test Loss: 0.7032 - Test Accuracy: 9.32% - Training DSC: 0.1809 -
EarlyStopping counter: 3 out of 5
Epoch 4 => Training Loss: 0.7010 - Training Accuracy: 10.07% - Test Loss: 0.6989 - Test Accuracy: 9.43% - Training DSC: 0.1809 -
Test IoU increased  0.126835 -->  0.126835. Saving model......
Epoch 5 => Training Loss: 0.6967 - Training Accuracy: 20.65% - Test Loss: 0.6946 - Test Accuracy: 45.07% - Training DSC: 0.1974
EarlyStopping counter: 1 out of 5
Epoch 6 => Training Loss: 0.6925 - Training Accuracy: 59.31% - Test Loss: 0.6903 - Test Accuracy: 67.16% - Training DSC: 0.2632
EarlyStopping counter: 2 out of 5
Epoch 7 => Training Loss: 0.6884 - Training Accuracy: 67.20% - Test Loss: 0.6861 - Test Accuracy: 70.10% - Training DSC: 0.1130
Test IoU increased  0.590197 -->  0.590197. Saving model......
Epoch 8 => Training Loss: 0.6844 - Training Accuracy: 84.34% - Test Loss: 0.6820 - Test Accuracy: 90.59% - Training DSC: 0.0117
Test IoU increased  0.657626 -->  0.657626. Saving model......
Epoch 9 => Training Loss: 0.6803 - Training Accuracy: 89.92% - Test Loss: 0.6779 - Test Accuracy: 90.67% - Training DSC: 0.0013
Test IoU increased  0.669653 -->  0.669653. Saving model......
Epoch 10 => Training Loss: 0.6764 - Training Accuracy: 89.94% - Test Loss: 0.6739 - Test Accuracy: 90.68% - Training DSC: 0.0001
Test IoU increased  0.669926 -->  0.669926. Saving model......
Epoch 11 => Training Loss: 0.6742 - Training Accuracy: 89.95% - Test Loss: 0.6735 - Test Accuracy: 90.68% - Training DSC: 0.0000
Test IoU increased  0.670255 -->  0.670255. Saving model......
Epoch 12 => Training Loss: 0.6738 - Training Accuracy: 89.95% - Test Loss: 0.6731 - Test Accuracy: 90.68% - Training DSC: 0.0000

The best check point is => Best Loss: 0.6697, Best Accuracy: 90.68%, Best DSC: 0.0000, Best IoU_score: 0.6712
```

V INDIVIDUAL CONTRIBUTIONS

Candidate 2023 contributes on:

- Responsible for the implementation of the segmentation task by selecting three different activation functions with a classical U-Net model, analysis and compare the different performance, alongside with discussions and discoveries.
- The candidate incorporates the augmentation technique to optimize the different activation functions-based model performance.
- Discover the saturating activation function does not perform well in the image segmentation task.
- Discover the image segmentation different result before and after data augmentation.
- Observe the learning track.
- The candidate takes responsibility for the data pre-processing.
- Participates in neural networks design: train and test model; the calculations of loss, accuracy, DSC, IoU; Early Stopping, computation of DSC and IoU, and visualization of images, ground truths and predictions.
- The candidate manages the thesis structure, composing of abstract and introduction part.

Candidate 2036 contributes on:

- The candidate participates in the data pre-processing, such as: importing the necessary libraries, downloading the dataset and making sure the images are transformed to the correct shape and size.
- Responsible for the implementation of the neural network models, architecture, and code structure for running the selected models U-Net, U-Net++ and FPN.
- In order to analyze and compare the different performance, alongside with discussions and discoveries, I made the preparations for my colleagues to implement their activation functions and optimizers.
- Participates in neural networks design: train and test model; the calculations of loss, accuracy, DSC, IoU, Early Stopping, computation of DSC and IoU, and visualization of images, ground truths and predictions and compare these with and without augmentation techniques to optimize the different models performance.
- The candidate participates to add more context to the thesis structure, findings and summary.

Candidate 2015 contributes on:

- Responsible for evaluating and refining three distinct optimizers for the U-Net model through testing and training.
- The candidate made discoveries that Stochastic Gradient Descent (SGD) is unsuitable for this particular dataset.
- The candidate participates in tweaking with various metrics such as loss, accuracy, DSC and IoU.
- The candidate participates in training the model with different hyperparameters such as the learning rate for increasing the accuracy.

REFERENCES

[1] Ali, Moez. 2023. “Introduction to Activation Functions in Neural Networks.” Data Camp. November 2023. https://www.datacamp.com/tutorial/introduction-to-activation-functions-in-neural-networks

[2] Bonnet, Alexandre. 2023. “Fine-Tuning Models: Hyperparameter Optimization.” Encord.com. August 23, 2023. https://encord.com/blog/fine-tuning-models-hyperparameter-optimization/

[3] Brownlee, Jason. 2023. “Activation Functions in PyTorch.” Machine Learning Mastery. May 3, 2023. https://machinelearningmastery.com/activation-functions-in-pytorch/

[4] Culurciello, Eugenio. 2017. “Neural Network Architectures.” Medium. Towards Data Science. March 23, 2017. https://towardsdatascience.com/neural-network-architectures-156e5bad51ba

[5] Ekman, Magnus. 2021. “Deep Learning Frameworks and Network Tweaks.” InformIT Database. August 17, 2021. https://www.informit.com/articles/article.aspx?p=3131594&seqNum=2

[6] Gandhi, Rohith. 2018. “A Look at Gradient Descent and RMSprop Optimizers.” Medium. June 19, 2018. https://towardsdatascience.com/a-look-at-gradient-descent-and-rmsprop-optimizers-f77d483ef08b

[7] Iakubovskii, Pavel. 2023. “Qubvel/Segmentation_models.pytorch.” GitHub. November 22, 2023. https://github.com/qubvel/segmentation_models.pytorch/tree/master

[8] Kingma, Diederik P, and Jimmy Ba. 2014. “Adam: A Method for Stochastic Optimization.” ArXiv.org. December 22, 2014. https://arxiv.org/abs/1412.6980

[9] Lee, Minhyeok. 2023. “Mathematical Analysis and Performance Evaluation of the GELU Activation Function in Deep Learning.” Journal of Mathematics 2023 (August): 1–13. https://doi.org/10.1155/2023/4229924

[10] Lindner, Holger A, Manfred Thiel, and Verena Schneider-Lindner. 2023. “Clinical Ground Truth in Machine Learning for Early Sepsis Diagnosis.” The Lancet Digital Health 5 (6): e338–39. https://doi.org/10.1016/S2589-7500(23)00070-5

[11] Liu, Liyuan, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2019. “On the Variance of the Adaptive

Learning Rate and Beyond." Arxiv.org, August. https://doi.org/10.48550/arXiv.1908.03265

[12] Maxime, Dev. "Image Segmentation W FPN (Segmentation_model)." n.d. Kaggle.com. Accessed November 22, 2023. https://www.kaggle.com/code/devmaxime/image-segmentation-w-fpn-segmentation-model

[13] Nomidl. 2022. "Difference between Sigmoid and Softmax Activation Function?" Nomidl. April 20, 2022. https://www.nomidl.com/deep-learning/what-is-the-difference-between-sigmoid-and-softmax-activation-function/

[14] qubvel. 2021. "Segmentation_models.pytorch/Examples/Binary_segmentation_intro.ipynb at Master · Qubvel/Segmentation_models.pytorch." GitHub. 2021. https://github.com/qubvel/segmentation_models.pytorch/blob/master/examples/binary_segmentation_intro.ipynb

[15] Rezatofighi, Hamid and Tsoi, Nathan and Gwak, JunYoung and Sadeghian, Amir and Reid, Ian and Savarese, Silvio, Generalized Intersection over Union. 2019 June. https://giou.stanford.edu/#method

[16] Sunde, Bjarte Mehus. 2022. "Early Stopping for PyTorch." GitHub. March 17, 2022. https://github.com/Bjarten/early-stopping-pytorch

[17] Thambawita, Vajira, Steven Hicks, Pål Halvorsen, and Michael Riegler. n.d. "DivergentNets: Medical Image Segmentation by Network Ensemble." Accessed November 22, 2023. https://ceur-ws.org/Vol-2886/paper3.pdf

[18] Yakubovskiy, Pavel "Tutorial — Segmentation Models 0.1.2 Documentation." n.d. Segmentation-Models.readthedocs.io. https://segmentation-models.readthedocs.io/en/latest/tutorial.html

[19] Zou KH, Warfield SK, Bharatha A, Tempany CM, Kaus MR, Haker SJ, Wells WM 3rd, Jolesz FA, Kikinis R. Statistical validation of image segmentation quality based on a spatial overlap index. Acad Radiol. 2004 Feb; https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1415224/

[20] "Pytorch Examples/Notebooks/PyTorch_Data_Augmentation_Image_Segmentation.ipynb at Master · Fabioperez/Pytorch-Examples." n.d. GitHub. Accessed November 22, 2023. https://github.com/fabioperez/pytorch-examples/blob/master/notebooks/PyTorch_Data_Augmentation_Image_Segmentation.ipynb

[21] "Torchvision.transforms — Torchvision Master Documentation." n.d. Pytorch.org. https://pytorch.org/vision/stable/transforms.html

[22] "Simula Datasets - Kvasir Instrument." n.d. Datasets.simula.no. Accessed November 22, 2023. https://datasets.simula.no/kvasir-instrument/