Integrating Active Contour Models with Attention Mechanisms in Deep Learning for Medical Image Segmentation

Anonymous ICCV submission

Paper ID 15581

Abstract

Medical image segmentation is essential for accurate diagnosis and treatment planning, particularly in identifying anomalies in CT and other imaging modalities. Our objective is to develop a deep learning-based segmentation framework that combines Active Contour Models (ACMs) and attention mechanisms, thus advancing automated medical image segmentation for improved clinical decision-making. By integrating ACMs with Convolutional Neural Networks (CNNs) and edge detection techniques, our approach enhances boundary precision and segmentation accuracy in medical images. Moreover, we investigate the novel idea of fine-tuning ACM hyperparameters by learning them within the CNN and backpropagating loss through the ACM. Our methodology addresses challenges such as weak boundaries, noise, and variability in anatomical structures, contributing to more robust and interpretable medical image analysis. This paper develops a prototype hybrid model and thoroughly evaluates it on multiple medical image datasets.

1. Introduction

Medical image segmentation is a critical task in medical diagnostics, enabling the precise delineation of anatomical structures and abnormalities in imaging modalities such as CT, MR, and ultrasound (US). However, segmentation remains challenging due to the complexity of medical images, which often exhibit irregular shapes, subtle boundaries, and significant noise. Additionally, effective segmentation often requires models trained on domain-specific datasets, making generalization difficult. Various approaches exist for medical image segmentation, each offering distinct strengths and limitations.

Deep learning-based segmentation models have gained prominence due to their ability to learn complex patterns from medical images. Notable architectures include U-Net and SegNet, which are widely used in medical image segmentation due to their encoder-decoder structures that help retain spatial information. Attention mechanisms, when integrated with CNNs, can further enhance the model's ability to focus on fine details, leading to more precise segmentation. The choice of loss function plays a crucial role in optimizing performance, with common functions such as Dice Loss, Intersection over Union (IoU), and Binary Cross-Entropy (BCE) tailored to handle class imbalance and segmentation accuracy.

Edge-based segmentation relies on detecting strong gradients to define object boundaries, but it may struggle with weak edges and noisy images. Recent vision-language models, such as the Segment Anything Model (SAM), leverage pre-trained transformers for zero-shot segmentation, though they may require fine-tuning for domain-specific tasks. Hybrid techniques combine traditional segmentation approaches with deep learning, offering improved accuracy, especially when dealing with weak boundaries or noisy data.

Active Contour Models (ACMs) offer an alternative approach to medical image segmentation by iteratively refining contours to fit object boundaries. ACMs provide several advantages, including adaptive boundary detection that dynamically adjusts contour points to capture complex and irregular shapes. They also allow for the integration of prior knowledge by fine-tuning energy terms based on domain-specific constraints, enhancing segmentation robustness. Unlike purely pixel-intensity-based methods, ACMs incorporate internal energy constraints, making them more resilient to image noise. Variants, such as region-based ACMs, further enhance flexibility. ACMs have proven versatile across various imaging modalities, including CT, MRI, and US, and are valuable for tasks such as organ boundary detection and tumor segmentation.

Our work introduces a novel approach to CNN-based segmentation for medical image datasets by integrating Active Contour Models (ACMs) to enhance boundary precision. A key innovation is the hybrid ACM model, which not only leverages pretrained edge attention mechanisms but also incorporates ACM logic directly into the CNN archi-

tecture. This allows the model to predict hyperparameters via an ACM hyperparameter generator as part of the training process, effectively setting these parameters dynamically rather than relying on manual tuning. By combining the strengths of deep learning and incorporating the DALS Level Set ACM [5], our approach aims to improve segmentation accuracy, robustness, and generalization across diverse medical images, marking a significant advancement in the field.

2. Related Work

Neural networks, particularly Convolutional Neural Networks (CNNs), have revolutionized the field of image processing and computer vision due to their ability to automatically learn spatial hierarchies of features from raw image data. CNNs are particularly effective in the domain of medical image segmentation because of their capacity to capture complex patterns in highly dimensional data, such as CT scans, MRI, and X-ray images.

U-Net, one of the most widely used architectures for medical image segmentation, is a CNN-based architecture specifically designed for biomedical image segmentation tasks. Ronneberger et al. [9] introduced U-Net as a fully convolutional network that is particularly effective for segmenting small and irregular structures in medical images.

Several variations of U-Net have been proposed in the literature to improve performance in specific medical imaging tasks, such as 3D U-Net for volumetric data developed by Cicek et al. [4], which extends the original U-Net for 3D image segmentation, making it well-suited for MRI and CT scans. The attention U-Net of Oktay et al. [8] introduces attention mechanisms to the U-Net architecture, allowing the model to focus on the most relevant regions of an image while suppressing irrelevant information, improving the segmentation quality in complex cases.

Active Contour Models (ACMs), introduced by Kass et al. [6] are deformable curves that evolve within an image to align with object boundaries by minimizing an energy functional. The model balances internal forces, which enforce smoothness, and external forces from image gradients that attract the curve to edges. ACMs have since been extended to handle topology changes, region-based energies, and deep learning integration.

Chan and Vese [3] proposed a region-based ACM that segments objects based on intensity homogeneity rather than edges. This model is more robust to weak or blurred edges and can segment multiple objects simultaneously, making it particularly useful in medical imaging and texture segmentation.

Certain popular hybrid frameworks utilize deep learning frameworks to get initial segmentation masks for images and later refine these segmentation masks using active contour models outside the learning framework. Hatamizadeh et al. [5] utilize a multiscale encoder-decoder CNN architecture whereas Nakhaei et al. [7] use the Segment Anything (SAM) architecture for the 'trained neural network' component in the image.

Vepa et al. [10] proposed a weakly supervised framework for cerebral vessel segmentation in DSA images. They use an active contour model to generate weak annotations, refined through human-in-the-loop strategies. These labels are fed into a CNN for further refinement. CLAHE is applied as a pre-processing step to enhance vessel visibility. The method reduces reliance on manual annotations and achieves state-of-the-art performance, surpassing human annotators.

3. Methodology

3.1. Convolutional Neural Network

3.1.1. Architecture

The network follows an encoder-bottleneck-decoder structure. The encoder extracts hierarchical features, incorporating attention mechanisms to suppress irrelevant information and refine feature selection. The bottleneck enhances feature representation and, if enabled, integrates an ACM Hyperparameter Generator to optimize contour refinement parameters. The decoder reconstructs the segmentation mask using transpose convolutions and attention layers, refining spatial details before outputting the final mask via a Sigmoid activation. Edge-based post-processing can further improve boundary accuracy.

By integrating attention mechanisms and ACMs, the Edge-Segmentation CNN achieves precise edge-based segmentation, making it well-suited for medical imaging applications.

3.1.2. Data Preprocessing

The data preprocessing pipeline for the edge segmentation model standardizes the dataset by processing grayscale images and their corresponding segmentation masks. Images are resized to 1024×1024 pixels, normalized to the $[0, 1]$ range, and converted into PyTorch tensors. A custom EdgeSegmentationDataset class is implemented to load and preprocess image-mask pairs, ensuring that only matching files are considered. The pipeline also supports optional transformations like data augmentation (random flips, rotations) and sample size limitations. This setup ensures consistency in input data, enhancing the model's stability and performance during training.

3.1.3. Attention Mechanisms

Attention mechanisms enhance the model's ability to focus on relevant features while suppressing irrelevant information. We compare two approaches: Base Attention and Edge Attention, highlighting their differences in feature refinement and edge awareness.

Our model supports Base and Edge Attention mechanisms. Base Attention refines feature importance, while Edge Attention enhances segmentation by incorporating edge detection using the Roberts Operator. The ACM module dynamically adjusts segmentation masks through contour-based optimization, ensuring smooth and well-defined boundaries.

Base Attention learns an attention map by processing the input feature map through a series of 1×1 convolutions, followed by a sigmoid activation to normalize attention values. The model scales the input features using these learned weights, refining feature importance. The process involves four key steps: feature transformation through a 1×1 convolution, refinement using another 1×1 convolution, normalization via sigmoid activation, and feature modulation, where the input features are multiplied by the attention map to selectively enhance relevant regions. However, Base Attention does not explicitly consider edge information, which can limit segmentation accuracy near boundaries. Since it applies attention uniformly across the feature space, it lacks specialized treatment of high-gradient regions.

Edge Attention extends Base Attention by explicitly incorporating edge information into the attention computation. It applies a separate edge-detection convolution, ensuring that attention focuses on boundary details to improve segmentation performance. This process starts with feature extraction using a 3×3 convolution, followed by edge awareness, where edges are extracted from the input. Then, a convolutional layer transforms edge features into attention weights, which are normalized via sigmoid activation. Finally, these computed attention weights are multiplied with the extracted features, enhancing segmentation precision around boundaries.

3.2. Active Contour Models

ACMs rely on several key components that collectively govern the contour evolution process. The intensity image serves as the primary input, representing the image on which the contour evolves. The initial contour defines the starting boundary, playing a critical role in the model's convergence and final segmentation outcome. The algorithm and energy functional dictate how the contour evolves over time, balancing internal forces for smoothness and external forces derived from the image. Additionally, hyperparameters control various aspects of the model's behavior, such as regularization strength and step size, significantly impacting the performance and stability of the segmentation.

We first describe the implementation of the DALS Level Set ACM and evaluate its performance with different initial contours and hyperparameters. Finally, we motivate the integration of this ACM into the CNN architecture, establishing the foundation for a unified segmentation framework.

3.2.1. Level Set ACM (LSA) Implemented in DALS

Since this is a well-defined implementation by Hatamizadeh et al. [5] related to the segmentation of medical images (particularly skin lesions), it serves as a good baseline for us to later incorporate into the CNN.

DALS Architecture An encoder-decoder CNN model is trained to produce good probability masks for segmenting images. The Level-Set ACM utilizes the CNN output probability mask and attempts to further perfect it to obtain a final probability mask.

Overview of the Level-Set ACM The Level-Set ACM begins with a 2D probability mask derived from the grayscale intensity image. First, $\lambda(x, y)$ maps are constructed based on predefined mathematical expressions. These maps assign weights to the contour energy terms, capturing intensity differences between pixels and the average intensity inside or outside the contour, depending on the pixel's location. This weighting helps guide the contour evolution process.

Next, an initial signed distance map is generated using the distance_transform_edt function from the scipy.ndimage library. The signed distance function $\phi(x, y)$ represents the shortest distance from each point to the closest point on the contour, with positive values inside the contour and negative values outside. The DALS LSA algorithm iteratively updates this map, evolving the contour until convergence. Finally, the resulting signed distance map is converted into a probability mask using the sigmoid function, producing the final segmentation result.

Level Set ACM Iteration Steps The main procedure for each energy minimization step is as follows:

- Narrow Band Identification: Pixels within a narrow band around the zero level set are selected as the region where computations are performed, improving computational efficiency.
- Intensity Estimation: The mean intensity values are computed separately for regions inside and outside the level set contour using local intensity information.
- Energy Terms Calculation: Two main energy terms are calculated based on intensity differences between the image and the estimated mean intensities inside and outside the contour. These terms encourage the contour to align with object boundaries where the intensity difference is maximized.
- Regularization Term: A curvature-based regularization term is computed to ensure smoothness of the evolving contour and prevent noisy or irregular boundaries. The regularization strength is controlled by the hyperparameter μ.

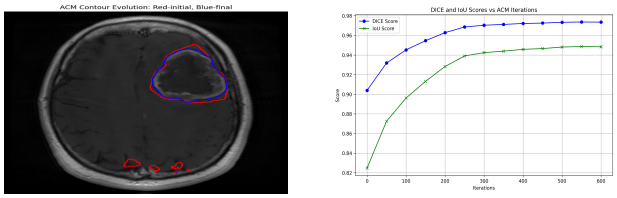

Figure 1. Experiment 1: DALS LSA Demo on brain image using a pretrained-CNN generated initial contour, $\nu = 5$, $\mu = 0.2$, number of iterations $= 600$

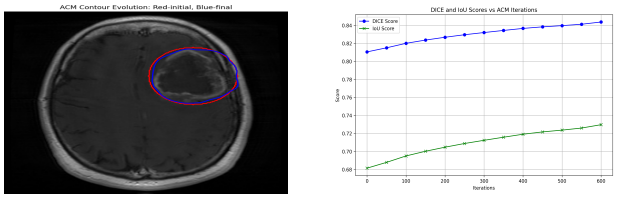

Figure 2. Experiment 2: DALS LSA Demo on brain image using a trivial circular initial contour, $\nu = 5$, $\mu = 0.2$, number of iterations $= 600$

- Balloon Force: An additional constant force, controlled by the hyperparameter ν, is applied to expand or shrink the contour, helping the contour evolve towards object boundaries.
- Force Computation: The total force guiding the contour evolution is calculated by combining the intensity-based energy terms, the curvature regularization term, and the balloon force.
- Level Set Update: The level set function is updated iteratively by applying the computed force, ensuring that the contour evolves toward object boundaries.
- Reinitialization: The level set function is periodically reinitialized to maintain numerical stability and preserve the signed distance function property.

The main hyperparameters for the Level-Set ACM are ν, μ, and the number of iterations.

Experiments of LSA on DALS brain demo Once we extracted and updated the DALS LSA, we were able to reproduce the DALS brain demo as shown in Figure 1. Here, we used the same initial contour (the one generated by pretrained CNN used in DALS) and hyperparameters ($\nu = 5$, $\mu = 0.2$, number of iterations $= 600$) specified in their demo. From the DICE/IOU score plots, one can infer that a larger number of iterations does not mean better results. Sometimes the improvement halts with more iterations or becomes worse. Hence, it is crucial to choose a good value for number of iterations.

In experiment 2 as shown in Figure 2, we used a little more trivial initial contour: a circle enclosing the area of interest. Now we see that the improvements are not nearly as significant as that in the previous experiment. A good initial segmentation allows the ACM to work well. The initial segmentation generated by a trained CNN is reasonably accurate with good probability mask values. This in turn leads to more accurate values for lambda and signed distance maps and hence better contour evolution. This experiment shows that DALS LSA is highly sensitive to initial contour.

In the next experiment we adhere to the original pretrained-CNN generated initial contour, and 600 iterations. The goal here is to initialize nu and mu to different value and see how this affects the DALS LSA performance.

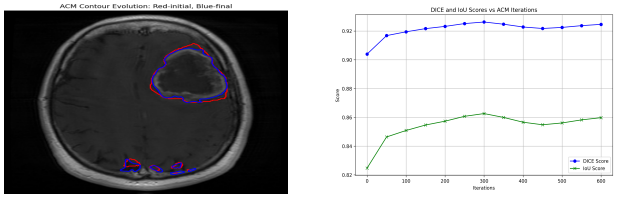

Figure 3. Experiment 3: DALS LSA Demo on brain image using a pretrained-CNN generated initial contour, $\nu = 0$, $\mu = 0$, number of iterations $= 600$

Setting both $\nu = \mu = 0$ leads to the results shown in Figure 3.

This experiment shows how critical the hyper-parameter setting is for an image. The wrong setting can lead to the ACM performance being very poor, while the correct setting leads to accurate results. Moreover, these settings depend on particular dataset and image characteristics. The current approach to setting these hyperparameters is manual trial and error which is infeasible.

DALS LSA tensorflow was time consuming and it was hard to simultaneously consider all possibilities and thoroughly experiment with different initial contours and hyperparameter settings. This motivates the need for hyperparameter setting and contour initialization to naturally be tuned through the neural network.

3.3. Overall Architecture

3.3.1. System Diagram

Figure 4 illustrates our hybrid ACM system. Enabling the ACM functionality in our architecture will include the "ACM Hyperparameter Generator," its outputs, and "Level Set ACM" of the above architecture, which will otherwise be disabled.

ACM Hyperparameter Generator The ACM Hyperparameter Generator uses adaptive average pooling as a flattening mechanism, followed by a fully connected layer with a ReLU activation and another fully connected layer with a Sigmoid activation. It produces three outputs: num_iters, ν, and μ, which are scaled to the correct ranges: 0–100 for num_iters, 0-10 for ν, and 0-1 for μ. The main novelty is

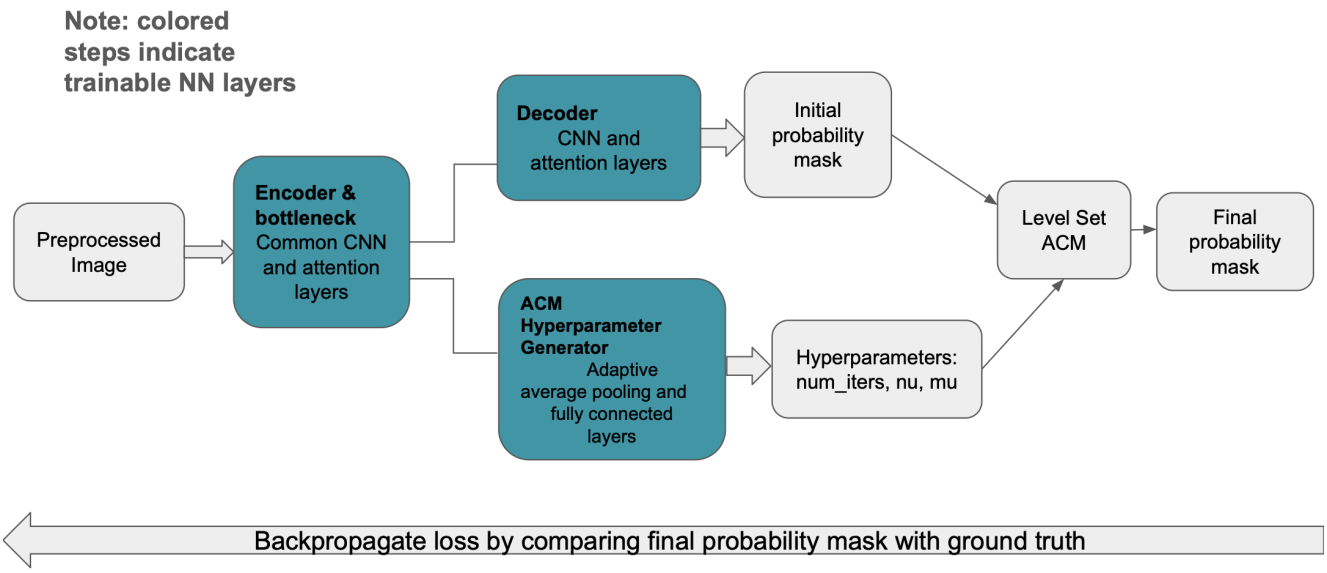

Figure 4. Hybrid ACM system diagram

to leverage the properties learned through the previous layers (encoder and bottleneck) and further use fully connected layers to learn image properties relevant to determining an optimal set of ACM hyperparameters that will result in effective ACM contour evolution.

Incorporating ACM into the Neural Network Incorporating the DALS Level Set ACM into the neural network required converting the original TensorFlow implementation to PyTorch, as our CNN baseline was in PyTorch. Additionally, the DALS LSA uses non-differentiable scipy functions to calculate the initial signed distance map from the probability mask. To address this, we aimed to replicate this logic using differentiable operations, ensuring compatibility with backpropagation. Making the DALS LSA fully differentiable will allow for more accurate gradient updates, improving the overall learning process. Figure 5 illustrates the differences in ACM contour evolution and demonstrates the success of our approach through the DALS brain demo results, showing that the converted DALS LSA logic remains intact. Furthermore, by working with a few images and epochs, we closely refined the code to eliminate backpropagation errors.

3.3.2. Training

The training procedure for the edge segmentation model follows a structured approach for efficient learning and generalization. The model is initialized, and its configuration is saved for reproducibility. Training is conducted in mini-batches, where predictions are made, and the loss is computed against ground truth masks. A learning rate scheduler adjusts the learning rate based on training progress, and

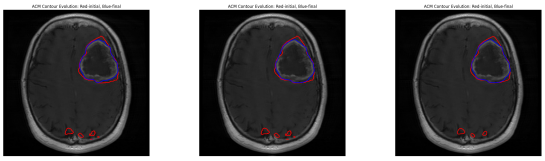

Figure 5. Demo of Tensorflow, Torch, and fully differentiable Torch versions of DALS Level Set ACM

gradient clipping prevents instability by controlling large gradients. Early stopping halts training if no improvement occurs after several epochs. Loss is logged and visualized to monitor progress, ensuring stable, efficient training and avoiding overfitting.

3.3.3. Testing

The testing process evaluates the trained model's performance on a test dataset. It begins by organizing the output folder based on the model's timestamp and epoch. The model is set to evaluation mode to ensure consistent behavior. Loss tracking and metric computation are initialized, and the model processes each batch in the test set to generate predictions. Loss is calculated, and predictions are converted to probabilities for metric evaluation. Visual results, including input-output-target comparisons, are saved. After processing all batches, performance metrics like accuracy and IoU are computed and stored. The results are saved for further analysis, providing insights into the model's effectiveness.

4. Results and Discussion

4.1. Considerations

Our model is trained with an initial learning rate of 0.001, which reduces by a factor of 0.5 if there is no improvement greater than 1e-4 after three epochs. The dataset used is the COVID-19 CT scan lesion segmentation dataset [1].

The paper discusses the use of different training configurations for various attention modules. The base attention module utilizes a model trained for 24 epochs, where convergence is achieved, and is trained on the entire COVID-19 CT scan dataset with an 80/20 train-test split ratio. The edge attention module uses a model trained for 14 epochs, where it begins to converge, also trained on the full dataset with the same train-test split. The hybrid ACM module, due to time complexity constraints, leverages the pretrained edge attention model (14 epochs) and is fine-tuned on a smaller subset of 16 training images and 4 testing images from the full dataset. The model implementation can be found in this GitHub repository [2].

4.2. Overall Results

In testing, the results are better and are produced at a faster rate for the Edge Attention model than the Base Attention model. While the loss is the same between the Base Attention and Edge Attention models, the number of epochs that were trained was much less. The resulting sample images, as will be seen in later sections, are much more adept at finding important segments in the CT-scans. While the Hybrid ACM model does have deficient loss compared to the Base and Edge Attention CNNs, its ability to have hyperparameter tuning for the ACM provide novel results and a foundation for future adaptation and training.

4.3. CNN with Base Attention

The results presented in Table 1 correspond to the CNN model trained with the base attention module. For a clearer comparison of the overall performance, please refer to Figure 6. While the average Dice Score of 0.3522 is respectable for a model with only a basic attention module, there is a significant disparity between the best and worst-case scenarios.

4.3.1. Results: Best vs Worst

The image that stands out as the best performer is shown in Figure 7. It achieved the highest precision among all the test images from the base attention model, with a Dice Score of 0.9590, which is an excellent result. While this is a strong performance, there is still potential for further improvement. Overall, these results highlight the model's strong capability and suggest opportunities for additional refinement.

The image that demonstrates the worst-case scenario is shown in Figure 8, with the corresponding results detailed in Figure 6. This image exhibited the lowest precision among all the test images from the base attention model, achieving a Dice Score of 0, indicating complete prediction failure in this instance. As illustrated in Figure 8, the final target image lacks sufficient detail, which likely confuses the model and impairs its ability to correctly identify and highlight the relevant features.

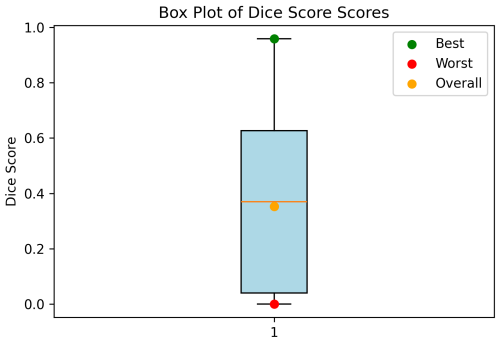

Figure 6. Dice Score Box Plot comparing results in the Base Attention Model

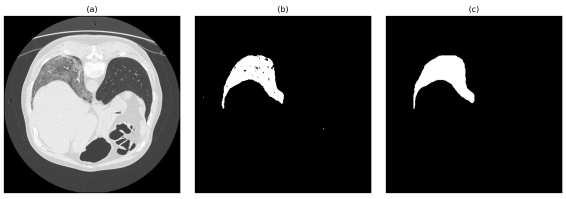

Figure 7. Best Base Attention Model result with a precision score of 0.9590 for the Jun_radiopaedia_7_85703_0_case20_39.png. (a) Input image in the dataset. (b) Predicted segmentation output from the trained model. (c) Target segmentation.

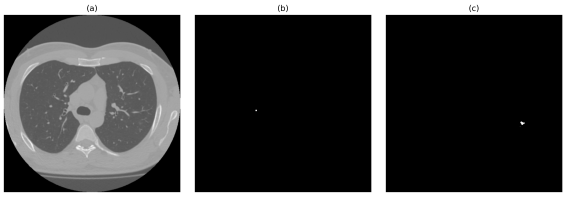

Figure 8. Worst Base Attention Model result with a Dice Score of 0.0 for the Morozov_study_0295_28.png. (a) Input image in the dataset. (b) Predicted segmentation output from the trained model. (c) Target segmentation.

4.4. CNN with Edge Attention

The results presented in Table 1 correspond to the CNN model trained with the edge attention module. For a clearer comparison of the overall performance, please refer to Figure 9. While the average Dice Score of 0.3580, which is not significantly better than the Base Attention model, it has

Metric	Base Attention Model	Edge Attention Model	Hybrid ACM Model
Average Test Loss	0.0316	0.0316	0.1691
AUROC	0.9801	0.9747	0.9512
AUC	0.9801	0.9747	0.9512
Precision	0.7377	0.8185	0.8341
Recall (Sensitivity)	0.4056	0.4253	0.8639
F1 Score	0.5235	0.5597	0.8487
IoU	0.2537	0.2710	0.7320
Dice Score	0.3522	0.3580	0.8451

Table 1. Performance Metrics for Base Attention Model (24 epochs trained), Edge Attention Model (14 epochs trained), and Hybrid ACM Model (10 epochs trained with 8 training images) Evaluation

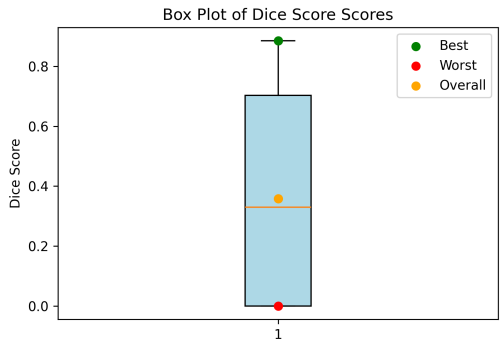

Figure 9. Dice Score Box Plot comparing results in the Edge Attention Model

much better results with less epochs trained which allows for faster convergence.

4.4.1. Results: Best vs Worst

An image to highlight is Figure 10 and its results are reported in Figure 9. This image had the best precision out of all the images tested from the edge attention model with a Dice Score of 0.8851. While this max result is not better than the Base Attention model, this result is from the model that has been trained for only 14 epochs, while the Base Attention model was trained for 24 epochs.

An image to highlight is Figure 11 and its results are reported in Figure 9. This image had the worst precision out of all the images tested from the edge attention model, with a Dice Score of 0.0. Similar to Figure 8, Figure 11 struggled to find the features to emphasize in segmentation, which is clear in its target result which is a very small segmentation.

4.5. Pretrained CNN with ACM

The main innovation of this approach is the ability to determine optimal ACM hyperparameters using fully connected layers, which significantly enhance the effectiveness of ACM contour evolution. Despite being trained on a small set of images atop the pretrained Edge Attention model, the results show a notable improvement in the performance of the ACM hyperparameter generator and the Hybrid ACM model. As shown in Table 1, the Hybrid ACM model achieves a Dice Score 2.4 times higher than the Base Attention model and 2.36 times higher than the Edge Attention model.

Figure 10. Best Edge Attention Model result with a Dice Score of 0.8851 for the Jun_coronacases_case9_137.png. (a) Input image in the dataset. (b) Predicted segmentation output from the trained model. (c) Target segmentation.

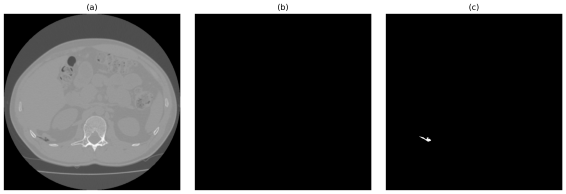

Figure 11. Worst Edge Attention Model result with a Dice Score of 0.0 for the Morozov_study_0288_3.png. (a) Input image in the dataset. (b) Predicted segmentation output from the trained model. (c) Target segmentation.

4.5.1. Results: Highlights

An image to highlight is Figure 12. This image had the best precision out of all the images tested from the Hybrid ACM model. The Dice Score is 0.8560, which is alongside

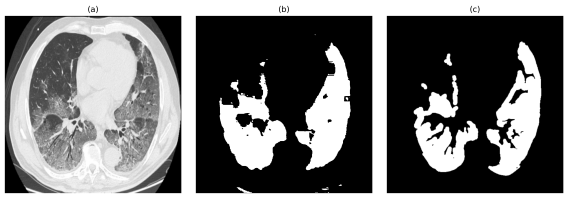

Figure 12. Hybrid ACM Model result with a Dice Score of 0.8560 for the bjorke_9.png. (a) Input image in the dataset. (b) Predicted segmentation output from the trained model. (c) Target segmentation.

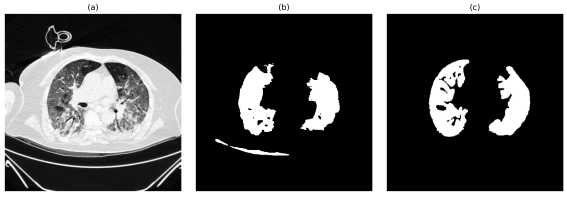

Figure 13. Hybrid ACM Model result with a Dice Score of 0.8343 for the bjorke_10.png. (a) Input image in the dataset. (b) Predicted segmentation output from the trained model. (c) Target segmentation.

the best performing result from the Edge Attention model, as seen in Figure 10 is acceptable. Despite both the Edge Attention and Hybrid ACM models best results performing with a lower Dice Score than the Base Attention model's best result, the Edge Attention was trained on 14 epochs and was used as the pretrained model for the Hybrid ACM. It provides grounds for faster convergence for a further trained Hybrid ACM model.

Another test result is Figure 13 and it had a similar Dice Score as its counterpart in Figure 12 of 0.8343 is significant, providing another result in the potential use of an ACM hyperparameter generator as a part of a Hybrid ACM model.

4.6. Discussion

The overall results from the Hybrid ACM model is novel in its approach to utilizing the ACM hyperparameter generator and provides a profound innovation in its possibility to learn future contours with a combined neural network and level-set ACM. The Active Contour Model (ACM) with hyperparameter tuning demonstrates excellent performance when trained and tested on small datasets. However, its application to larger datasets presents significant computational challenges due to the high time and memory consumption associated with gradient tracking over hundreds of ACM iterations. Since each iteration contributes to refining the segmentation mask, the backpropagation process becomes increasingly expensive, leading to inefficiencies in training. This limitation makes large-scale training impractical without substantial computational resources. Despite this, the model's strong results on small datasets highlight its potential effectiveness, suggesting that it is well-suited for applications where high-quality segmentation is required on limited data samples. Future optimizations could focus on reducing computational overhead while preserving the model's ability to learn optimal ACM hyperparameters.

5. Conclusion

We have proposed a deep learning-based segmentation framework that combines Active Contour Models (ACMs) and attention mechanisms. The novelty of our approach lies in integrating edge segmentation with a CNN, enhancing feature extraction by emphasizing structural boundaries. Incorporating the ACM as an additional component refines the segmentation process by leveraging its ability to capture object contours dynamically. Moreover, our main novelty is introducing an "ACM Hyperparameter Generator" into the image segmentation CNN that can be trained to generate image-dependent optimal ACM hyperparameters for effective segmentation refinement. While the loss remains high for a period during training, this hybrid approach balances data-driven feature learning with contour-based refinement, potentially improving segmentation accuracy over time.

Future work will focus on several key areas for improvement. This includes exploring additional image transformations and pre-processing techniques to enhance model performance. Further training will be conducted on a broader range of variations to increase the model's robustness. Additionally, model improvements such as incorporating batch normalization will be explored, along with optimizing the ACM to enhance speed and memory efficiency. A sliding window approach will also be investigated for improving the handling of large images during training. Finally, further training will be conducted with the optimized ACM to refine its performance.

References

[1] Covid-19. http://medicalsegmentation.com/covid19/, 2020. Accessed: 23 December, 2020. 6

[2] Shairaalam Alam. Medical image segmentation, 2025. Accessed: 2025-03-07. 6

[3] T. F. Chan and L. A. Vese. An active contour model without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001. 2

[4] O. Cicek, A. Abdulkadir, S. S. Lienkamp, and et al. 3d u-net: Learning dense volumetric segmentation from sparse annotation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 424–432, 2016. 2

[5] A. Hatamizadeh, A. Hoogi, D. Sengupta, W. Lu, B. Wilcox, D. Rubin, and D. Terzopoulos. Deep active lesion segmentation. In *Machine Learning in Medical Imaging*, pages 98–106, 2019. 2, 3

[6] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active

contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988. 2

[7] N. Nakhaei, T. Zhang, D. Terzopoulos, and W. Hsu. Refining boundaries of the segment anything model in medical images using an active contour model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 2

[8] O. Oktay, J. Schlemper, L. L. Folgoc, and et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint*, 2018. 2

[9] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, 2015. 2

[10] A. Vepa, A. Choi, N. Nakhaei, W. Lee, N. Stier, A. Vu, G. Jenkins, X. Yang, M. Shergill, M. Desphy, K. Delao, M. Levy, C. Garduno, L. Nelson, W. Liu, F. Hung, and F. Scalzo. Weakly-supervised convolutional neural networks for vessel segmentation in cerebral angiography. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 585–594, 2022. 2