



Cyprus
University of
Technology



杭州电子科技大学
HANGZHOU DIANZI UNIVERSITY

Master's Thesis

**Enhanced Automated Prostate Segmentation in Ultrasound
Images Based on Diverse Pre-Processing Strategies and Multi-
Input Architectures**

Jiale Hou

Limassol, May 2025



**MSc in Electronics Science
and Technology**

CYPRUS UNIVERSITY OF TECHNOLOGY
FACULTY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER
ENGINEERING AND INFORMATICS

Master's Thesis

Enhanced Automated Prostate Segmentation in Ultrasound
Images Based on Diverse Pre-Processing Strategies and Multi-
Input Architectures

Jiale Hou

Supervisor

Prof. Christos P. Loizou

Limassol, Month and year of thesis submission

Approval Form

Master's Thesis

Enhanced Automated Prostate Segmentation in Ultrasound Images Based on Diverse Pre-Processing Strategies and Multi-Input Architectures

Presented by

Jiale Hou

Supervisor: [Prof. Christos P. Loizou]

Member of the committee: [Haohan Yu]

Member of the committee: [Efthymoulos Kyriakou]

Cyprus University of Technology

Limassol, Month and year of thesis submission

Copyrights

Copyright© [2025] [Jiale Hou]

All rights reserved.

The approval of the thesis by the Department of Electrical Engineering and Computer Engineering and Informatics does not necessarily imply the approval by the Department of the writer's views.

Acknowledgements.

I would like to express my deepest gratitude to Prof. Christos P. Loizou and Prof. Xiwei Huang for their invaluable guidance, insightful suggestions, and continuous support throughout the course of this research. Their expertise and mentorship have been instrumental in shaping both the direction and the quality of this work.

My sincere thanks also go to the team at HealthLab, Cyprus University of Technology, for providing a highly collaborative and intellectually stimulating research environment. In particular, I am deeply grateful to PhD Georgia D. Liapi, whose detailed feedback, technical advice, and constant encouragement significantly improved the outcome of this study.

I would also like to acknowledge the German Oncology Center in Nicosia, Cyprus, for providing the medical imaging data used in this work. Their contribution was critical in supporting the practical application and clinical relevance of the proposed segmentation framework.

ABSTRACT

This study presents a comprehensive framework for enhancing automated prostate segmentation in transrectal ultrasound (TRUS) images using a combination of tailored preprocessing techniques, dual-input architectures, and automated hyperparameter optimization. TRUS images are widely used in clinical practice but suffer from low contrast, speckle noise, and boundary ambiguity, which present challenges for accurate segmentation. To address these issues, this work evaluates three preprocessing schemes—intensity normalization, despeckling, and their combination—to improve input consistency and reduce imaging artifacts. Furthermore, it introduces and compares two multi-input strategies: channel expansion and a dual-branch network that separately processes normalized images and structural priors (binary or contour maps). The segmentation is performed using a modified DeepLabv3+ architecture with a MobileNetV2 backbone, and hyperparameters are optimized through a two-stage grid and Bayesian search. Experimental results on 289 TRUS images demonstrate that the dual-branch architecture, particularly the NBD strategy, achieves superior performance, with a Dice coefficient of 0.969 and strong robustness across varying conditions. These results confirm the framework’s effectiveness in enhancing accuracy and generalizability, supporting its potential clinical utility.

Keywords: Prostate segmentation, Ultrasound imaging, DeepLabv3+, Dual-input architecture, Hyperparameter optimization

Table of Contents

ABSTRACT.....	vi
LIST OF TABLES.....	ix
LIST OF FIGURES.....	x
LIST OF ABBREVIATIONS.....	xi
1 Introduction.....	1
2 Literature review.....	
3 Research Methodology.....	1
3.1 Acquisition of Ultrasound Images and Manual Delineations.....	10
3.2 Image Pre-processing.....	12
3.2.1 Image Intensity Normalization.....	12
3.2.2 Despeckling.....	14
3.3 Segmentation Algorithm.....	16
3.3.1 DeepLabv3+ architecture.....	16
3.3.2 Muti-Input.....	18
3.3.3 Automatic Hyperparameter Optimization Algorithm.....	20
3.3.4 Early stop.....	22
3.3.5 Evaluation Metrics for the Segmentation.....	23
4 Results.....	25
5 Discussion.....	40
5.1 Pre-Processing.....	41
5.2 Fusion strategy.....	45
5.3 Advantages and disadvantages of the proposed segmentation method.....	49
5.4 Future directions.....	49
6 Conclusions.....	54

BIBLIOGRAPHY	56
APPENDIX I	63

LIST OF TABLES

Table 2-1 Summary of deep learning (DL)-based prostate segmentation studies on TRUS images from the current literature.	8
Table 3-1: Modified MobileNetV2 Architecture Used as the Backbone in the Proposed Network [12].....	18
Table 4-1: Statistical summary of segmentation metrics across four preprocessing methods on the whole test set of PCa images investigated in this study (N=58)	29
Table 4-2: Statistical summary of segmentation metrics comparing normalized input (N) with different dual-input strategies (NBD, NBE, NCD, NCE) on the whole test set of PCa images investigated in this study (N = 58).....	33
Table 4-3: Statistical summary of segmentation metrics comparing normalized input (N) with different dual-input strategies (NBD, NBE, NCD, NCE) on the whole test set of PCa images investigated in this study (N = 58).....	38

LIST OF FIGURES

Figure 3-1: Flow Diagram of the proposed DL prostate image segmentation method ..	11
Figure 3-2: Original (O) and normalized (N) ultrasound images of the prostate with manually selected reference regions for intensity normalization	12
Figure 3-3: Comparison of ROI Pixel Intensity Histograms Before and After Normalization	13
Figure 3-4: Heatmap of a 2D Gaussian Kernel Used for Image Smoothing	14
Figure 3-5: Comparison of Original (O) and Despeckled (D) Ultrasound Images	15
Figure 3-6: Architecture of the Proposed Semantic Segmentation Network.....	16
Figure 3-7: Structure of the Basic Building Blocks in MobileNetV2 [ref]	17
Figure 3-8: Illustration of Pre-processing Steps: Normalization (N), Binarization (B), and Initial Contour Extraction (C) in an ultrasound prostate image.....	19
Figure 3-9: Channel Expansion and Fusion Prior to Backbone Input	20
Figure 3-10: Overall Architecture of the Proposed Network with Dual Input and Contextual Feature Fusion	21
Figure 3-11: Illustration of the Automatic Hyperparameter Optimization Method	22
Figure 4-1: Segmentation performance comparison across different preprocessing strategies on clear (top row) and ambiguous (bottom row) prostate boundaries.....	26
Figure 4-2: Box plots of segmentation EM for N PCa images	27
Figure 4-3: Bland-Altman plot in Fig. 4-3a) and linear regression analysis in Fig. 4-3b), of the DC under different preprocessing methods (N vs D)	28
Figure 4-4 Enhanced segmentation results under different preprocessing and input fusion strategies following automated hyperparameter optimization.....	31
Figure 4-5: Box plots of the DC for different input fusion strategies on images investigated in this study (N=58). The (median, IQR) are shown below each boxplot..	34
Figure 4-6: Bland-Altman plot (see Fig. 4-6a) and correlation analysis of the DC (see Fig. 4-6b) between the N and NBD groups on images investigated in this study.....	35

LIST OF ABBREVIATIONS

PCa:	Prostate Cancer
TRUS:	Transrectal Ultrasound
DL:	Deep Learning
N	Intensity Normalization
D	Despeckling
ND	Normalized + Despeckled
B	Binarization
C	Initial Contour
NBE	Normalized + Binary, Expanded Channel
NBD	Normalized + Binary, Dual-Branch
NCE	Normalized + Contour, Expanded Channel
NCD	Normalized + Contour, Dual-Branch
SDENN	Storage-based Differential Evolution Neural Networks
ConvGRU	Convolutional Gated Recurrent Units
ROI	Region of Interest
EM	Evaluation Metric(s)
DSC	Dice Similarity Coefficient
AC	Accuracy
P	Precision
S	Specificity
CK	Cohen's Kappa
HD	Hausdorff Distance
SD	Standard Deviation
ρ	Spearman Rank Correlation Coefficient

1 Introduction

1.1 Background and Motivation

Prostate cancer (PCa) is one of the most prevalent malignancies affecting men worldwide and is a leading cause of cancer-related mortality. With an aging population, the demand for early diagnosis and accurate treatment of PCa continues to grow [1]. Among various imaging modalities, Transrectal Ultrasound (TRUS) stands out for its cost-effectiveness, non-invasiveness, and real-time imaging capabilities, making it widely used in clinical prostate examination. However, TRUS images suffer from challenges such as speckle noise, low tissue contrast, intensity inhomogeneity, and highly variable prostate shapes and boundaries. These limitations hinder the performance of automated segmentation algorithms and restrict their clinical deployment. Automated prostate segmentation is crucial for biopsy guidance, treatment planning, and longitudinal disease monitoring [2].

In recent years, deep learning (DL)-based segmentation models have shown tremendous promise in tackling medical image analysis tasks. Encoder-decoder architectures such as U-Net and DeepLabv3+ have achieved state-of-the-art performance in various segmentation challenges [3]. Nevertheless, when applied to TRUS images, these models often struggle, especially with images that exhibit blurry or ambiguous prostate boundaries. The primary obstacles include (1) image appearance variability due to acquisition settings (e.g., gain and echogenicity differences), and (2) a lack of explicit structural guidance during segmentation [4].

1.2 Innovation

This research centers on enhancing prostate segmentation accuracy and robustness on TRUS images through a systematic investigation of image preprocessing strategies and dual-input guidance mechanisms. Specifically, we explore three preprocessing approaches: intensity normalization (N) [5], despeckling (D)[6], and a combination of both (ND). These techniques aim to reduce inter-image variability and provide consistent intensity representations for the model.

To further guide the segmentation network, we design two structural guidance strategies: channel expansion and dual-branch input fusion. Each strategy incorporates additional

structural cues—either binary maps or initial contour maps—alongside the preprocessed images. In the channel expansion approach, the auxiliary image is concatenated with the normalized image along the channel dimension. In contrast, the dual-branch structure processes the two inputs independently through separate branches before merging their features [7].

This study covers the full pipeline from preprocessing design and structural input formulation to in-depth evaluation across different image quality conditions. By experimenting with combinations such as NBE, NBD, NCE, and NCD, we assess the effectiveness of each preprocessing and fusion method.

1.3 Objectives

The central objective of this study is to design and validate an enhanced deep learning-based segmentation approach for transrectal ultrasound (TRUS) prostate images that effectively handles the inherent challenges associated with this imaging modality. These challenges include high levels of speckle noise, low contrast, intensity inhomogeneity, and significant variability in anatomical structure, especially under poor image quality conditions. Traditional segmentation methods often struggle under these conditions, leading to inconsistent and unreliable clinical outcomes [8]. Therefore, this study aims to create a segmentation framework that not only improves segmentation accuracy but also exhibits robustness across diverse image qualities.

To this end, the study first investigates the role of image preprocessing in enhancing segmentation performance. Three preprocessing schemes are evaluated: (1) intensity normalization (N) [5], which aims to standardize image brightness and contrast; (2) despeckling (D) [6], which reduces noise while preserving structural features; and (3) a combined approach (ND)[6] that integrates both normalization and despeckling. These techniques are intended to improve the quality of input data and facilitate more accurate learning by the network.

The study then focuses on the design and evaluation of dual-input segmentation strategies. Two integration strategies are proposed and compared: the channel expansion method and the dual-branch architecture [7]. Both strategies use an additional guidance input—either a binarized mask or an initial contour map derived from the original image—to provide structural context during segmentation. The dual-branch architecture is specifically designed to independently process the two input channels before fusing their features, with the aim of leveraging complementary information more effectively.

Additionally, the study addresses the often-overlooked aspect of training configuration by incorporating an automatic hyperparameter optimization mechanism [9]. This two-stage strategy first employs a grid search to determine the best combination of discrete

hyperparameters (batch size, optimizer type) and then applies Bayesian optimization to fine-tune continuous parameters [10], particularly the learning rate. The objective is to enhance convergence speed, model accuracy, and generalization capacity.

Collectively, this study seeks to answer the following research questions:

1. Which preprocessing strategy yields the most consistent and accurate segmentation results for TRUS images.
2. How do different dual-input integration methods impact segmentation performance, particularly in images with ambiguous boundaries.
3. To what extent can automated hyperparameter tuning improve segmentation accuracy and training efficiency.

By answering these questions, this research aims to provide a clinically relevant, technically robust framework for automatic prostate segmentation in ultrasound images.

1.4 Contributions

This research makes several novel and practical contributions to the field of medical image segmentation, particularly in the context of prostate cancer diagnosis and treatment planning using TRUS imaging. The key contributions are as follows:

- **Systematic Evaluation of Preprocessing Strategies:** The study presents a thorough analysis of how different preprocessing methods affect segmentation outcomes [5], [6]. By comparing intensity normalization, despeckling, and their combination, the study identifies preprocessing configurations that are particularly effective in enhancing image clarity and aiding model learning.
- **Development and Assessment of Dual-Input Architectures:** Two dual-input fusion strategies are proposed—channel expansion and dual-branch. The dual-branch model, which processes each input independently before feature fusion, is shown to outperform simpler fusion methods [7]. This architecture is especially effective in segmenting images with poor boundary definition, demonstrating the value of incorporating structural priors such as binary maps and contour images.
- **Incorporation of Structural Guidance:** The study demonstrates that supplying structural priors (e.g., binary masks or initial contours) to the segmentation model can significantly enhance boundary delineation [11]. This is particularly useful in clinical scenarios where the prostate boundaries are not clearly visible due to image quality limitations.
- **Automated Hyperparameter Optimization:** By integrating grid search with Bayesian optimization, the study introduces an efficient and systematic approach to hyperparameter tuning [10]. This contributes to improved model performance without the need for exhaustive manual experimentation.
- **Comprehensive Evaluation and Statistical Validation:** The proposed methods are validated using a broad set of evaluation metrics, including Dice coefficient (DC), precision (P), specificity (S), accuracy (AC), Cohen’s Kappa (CK), and Hausdorff Distance (HD) [12]. Additionally, statistical analyses such as the Wilcoxon rank-sum test, Bland-Altman analysis, and Spearman correlation are employed to ensure the reliability and significance of the results [13].
- **Performance Benchmarking Against Literature:** The study situates its findings within the broader research context by comparing results with those reported in recent literature. It highlights how the proposed dual-input strategies and preprocessing methods achieve competitive or superior results in terms of segmentation accuracy and robustness.
- **Real-World Clinical Applicability:** The proposed framework demonstrates strong performance not only on high-quality images but also in cases with poor boundary visibility. This robustness suggests that the method could be effectively

integrated into real-world clinical workflows, assisting radiologists in accurately identifying prostate boundaries during diagnosis and treatment planning.

- **Scalable and Flexible Design:** The modular structure of the proposed framework allows for easy adaptation to other medical imaging modalities and organs, laying the groundwork for future research in multi-organ segmentation tasks.

In summary, this study delivers a comprehensive, flexible, and clinically relevant solution to the problem of prostate segmentation in TRUS images by combining methodological innovation, rigorous validation, and practical design.

2 Literature review

Several studies have explored deep learning-based prostate region segmentation methods using TRUS images (Table 1). Van Sloun et al.[14], [15] deeply integrated advanced technologies to construct an efficient segmentation architecture, innovating the fully convolutional neural network based on U-Net. At the data processing end, histogram equalization was used to enhance image features, normalization was applied to unify data scales, and multi-strategy data augmentation, including affine and elastic deformations, was employed to enhance the model's generalization and robustness. For optimization, the Adam optimizer was introduced to minimize cross-entropy loss, driving the model towards accurate convergence. The unique network architecture, such as the encoder-decoder module with skip connections, effectively fused multi-scale features.

Anas et al.[2] expanded on the U-Net architecture, using dilated convolutions to improve local predictions and residual convolutions to optimize training. For feature extraction, convolutional gated recurrent units (ConvGRU) were used to mine temporal features from ultrasound images, with GRU serving as the core, and update and reset gates modulating the feature flow. Residual convolutions were used for optimization. Innovative cyclic connections were employed, with ConvGRU replacing residual convolutions for enhancement; specific layers embedded ConvGRU and interconnected branches, strengthening feature fusion to improve segmentation. Additionally, far-frame information was also included with different frame sequences and parameters to enhance the network's robustness against ultrasound artifacts, achieving accurate segmentation even under calcification shadows.

Karimi et al.[3] focused on improving the accuracy and robustness of prostate CTV segmentation in TRUS images. They used an adaptive sampling strategy, analyzing

training image features with convolutional autoencoders and clustering them, then selecting samples based on their similarity to difficult-to-segment images in the validation set, helping the model focus on challenging images and improving generalization. A CNN ensemble model was constructed to assess segmentation uncertainty, using a unique algorithm to generate uncertainty maps. To address uncertain segmentations, a statistical shape model was introduced, combining prior shape knowledge and uncertainty maps to optimize boundaries.

Vesal et al.[16] innovatively built a Coordinated Dilated Residual UNet (CoordDR-UNet) network architecture, using 2.5D input to integrate multi-scale information, dilated convolutions to expand the receptive field, residual connections to optimize gradient flow, and coordinate attention modules to focus on key regions, improving accuracy, especially in enhancing the segmentation of fuzzy prostate boundaries. To address the domain generalization problem in multi-institutional data, they proposed a knowledge distillation loss-based model training strategy, maintaining the model's knowledge of the source domain during fine-tuning.

Peng et al.[17] first used an attention-gated U-Net for rough segmentation of the prostate's general contour, then refined it using optimized multi-segment tracking and storage-based differential evolution neural networks (SDENN). They improved the OPST method by optimizing normalization, projection, vertex optimization, and cleaning steps, with Z-score normalization for robustness, local scanning for efficiency, new functions for vertex optimization, and rule-based filtering of outliers to accurately refine the contour. In SDENN, SDE uses stored reuse, multi-operators, and dynamic scaling for optimal global search, while ABNN selects suitable functions and parameters to solve gradient challenges for accurate boundary fitting.

Jiang et al.[18] proposed the MicroSegNet model for prostate micro-ultrasound image segmentation. Its core innovation lies in the fusion of a unique loss function and multi-scale deep supervision modules. The newly designed AG-BCE loss function cleverly defines the difficulty of segmenting regions based on expert and non-expert annotations, applying heavy penalties to errors in difficult regions, guiding the model to focus on learning features from critical areas, thus enhancing its ability to segment complex boundaries. The multi-scale deep supervision module embeds multi-scale convolutional layers at intermediate network layers, generating multi-resolution segmentation images

and incorporating differences between predicted and corresponding downsampled ground truth into the loss calculation, effectively combining high and low-level information and improving the model's perception of both global context and local details.

As shown in **Table 2-1**, most of the studies above have achieved significant model performance in prostate region segmentation. However, many of the proposed systems lack effective and detailed preprocessing steps for images, which are beneficial for prostate region segmentation. Furthermore, these methods primarily rely on simple min-max normalization[5] and contrast enhancement, without targeted normalization and denoising[6] for ultrasound image artifacts and noise. In contrast to the aforementioned studies, we propose and evaluate three preprocessing strategies for TRUS images, including intensity normalization, Gaussian filtering to remove noise, and the combined application of intensity normalization and Gaussian filtering, as recommended preprocessing steps before deep learning-based prostate region segmentation [citation needed]. Importantly, we focus on:

- Applying DeepLabV3+ for prostate ultrasound image segmentation, training on prostate ultrasound images with the aim of improving the accuracy of prostate region segmentation [19].
- Evaluating the impact of image preprocessing on the overall performance of the model in prostate region segmentation, as well as the evaluation of plaque type segmentation [12], [13].

Although the studies reviewed in **Table 2-1** have contributed significantly to the field of prostate ultrasound image segmentation, the method proposed in this study introduces several key innovations that distinguish it from previous approaches. Firstly, unlike most existing methods that rely solely on standard normalization or basic image enhancement, this work systematically explores three distinct preprocessing strategies—intensity normalization, despeckling filtering, and their combination [20]. These methods effectively reduce noise and enhance structural clarity in TRUS images, thus providing a more reliable input foundation for deep learning models.

Secondly, in terms of input architecture, prior works generally employ single-channel inputs without incorporating structural guidance. In contrast, this study proposes two dual-input strategies: channel expansion and dual-branch design.[7] By incorporating

Table 2-1 Summary of deep learning (DL)-based prostate segmentation studies on TRUS images from the current literature.

Author	Year	NP/NI	PP	Model	TIS	SRTT [%]	N	DC[mean±std] d[%]
Van Sloun <i>et al.</i> [5]	2018	78/210	GIN, AHE	UNet	-	75-25	48	96±2.6
Anas <i>et al.</i> [6]	2018	18/29394	-	UNet, CGRU	224×224	75-25	1017	92.9±2.7
Karimi <i>et al.</i> [7]	2019	675/6750~	-	UNet	415×490	5KF	1350	93.9±3.5
Vesal <i>et al.</i> [8]	2022	2067/2920	GIN, AHE	UNet, DC, CAB	128×160	80-20	220	91.0±3.0
Peng <i>et al.</i> [4]	2023	300/1250	None	UNet, AGM	600×450	75-25	300	94.2±3.2
Jiang <i>et al.</i> [19]	2024	75/2818	GIN	TransUNet, MDSM	224×224	75-25	758	93.90
This study	2024	9/289	N, D, ND	DeepLabv3+	512×512	80-20	58	94.2±2.4

NP: Number of patients, NI: Number of images, PP: Preprocessing, TIS: Training input size, SRTT: Split rule training gest, N: Evaluated images, DC: Dice coefficient (mean±std), GIN: Global intensity normalization [13, 15], AHE: Adaptive histogram equalization [20], N: Intensity normalization [14, 15], D: Despeckle filtering [12], ND: Normalization and despeckle filtering N: Intensity normalization [14], DC: Dilated convolution [21], CAB: Coordinate attention block [16], MDSM: Multi-scale deep supervision module [22], AGM: Attention gate module [10], CGRU: Convolutional gate recurrent unit [16].

binary maps or initial contour maps as auxiliary inputs, the model benefits from additional spatial and structural information. Notably, the dual-branch structure allows separate encoding of the two input streams before fusing their features, significantly improving segmentation performance in regions with ambiguous boundaries—an area where conventional models tend to struggle [21].

Lastly, while most literature uses fixed or manually tuned hyperparameters, this research integrates an automated hyperparameter optimization mechanism. By combining grid search for discrete variables and Bayesian optimization for continuous ones, the training process becomes both more efficient and better adapted to the dataset, enhancing overall model robustness and generalization [10].

In summary, the proposed method introduces novel preprocessing pipelines, dual-input designs, and an automated parameter tuning strategy. These innovations collectively offer improvements in segmentation accuracy, stability, and clinical applicability, addressing important limitations in the current body of literature.

3 Research Methodology

In this Chapter the methodology of the proposed system is presented based on the flow diagram of Fig. 3.1, which is separated in the following steps: 1. data acquisition, 2. Data selection, 3. data preprocessing, 4, model input, 5. Automated segmentation, 6. Model output, 7. Segmentation visualization and overlay, 8. Evaluation, 9. binarize and contour detection, 10. Muti input, 11. Evaluation.

3.1 Acquisition of Ultrasound Images and Manual Delineations

We included a total of 647 transrectal ultrasound (TRUS) images of the prostate from nine patients, collected at the German Oncology Center (GOC) in Limassol, Cyprus. These images were acquired using a BK Medical ultrasound system (model BK3000), equipped with an E14CL4b (9048) endocavity biplane transducer [22]. This specialized probe is designed for high-resolution internal imaging and is particularly well-suited for prostate examinations. Image acquisition was carried out using B-mode scanning, which provides grayscale imaging of tissue structures, in combination with transverse scanning techniques to capture cross-sectional views of the prostate [23].

The imaging protocol consisted of obtaining a series of transverse images, beginning from the cranial end of the prostate and progressing systematically toward the caudal direction. This approach ensured thorough anatomical coverage of the prostate gland. All images were acquired without any needle insertion, preserving the anatomical integrity of the scanned area. It is important to note that not all image slices from each patient contained visible prostate tissue [24]. Therefore, after reviewing the full dataset and excluding images that did not contain relevant anatomical regions, we retained a final subset of 289 TRUS images across all nine patients (see Figure 3-1, step 2).

All imaging and associated clinical data were collected following a written informed consent from the patients and were anonymized prior to use. For the purpose of establishing ground truth (GT) segmentations, a radiation oncologist manually delineated the boundaries of the prostate region in selected images beyond image N [25] (see Figure 3-1, step 1), providing accurate reference annotations for further analysis and model training.

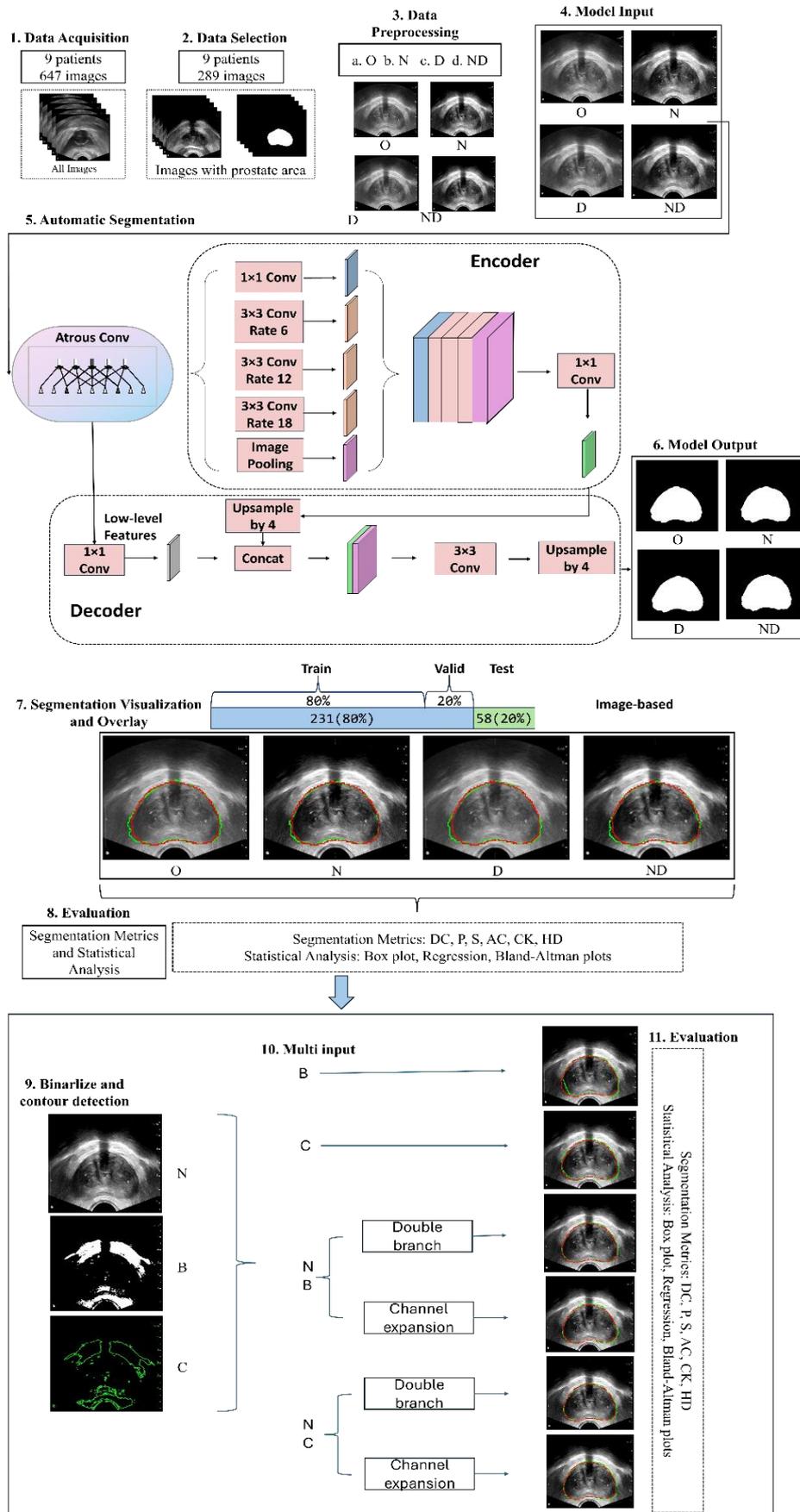


Figure 3-1: Flow Diagram of the proposed DL prostate image segmentation method

3.2 Image Pre-processing

3.2.1 Image Intensity Normalization

In medical image analysis, particularly in the domain of semantic segmentation, intensity normalization plays a pivotal role in enhancing the robustness and generalization capability of deep learning models. Variations in intensity distributions across images—caused by differences in patient anatomy, acquisition settings, or imaging protocols—can introduce inconsistencies that hinder model performance [26]. Therefore, implementing a tailored normalization strategy is essential to reduce such inter-image variability and ensure consistent data representation for network training.

In this study, we focus on transrectal ultrasound (TRUS) images, which differ from other modalities such as CT or MRI in terms of grayscale intensity characteristics. Unlike CT or MRI, where pixel values are often spread across a broad and well-defined intensity range (e.g., from 0 for air to 255 for bone), TRUS images typically do not contain pure black (0) or pure white (255) regions, and their effective intensity range can vary significantly from image to image due to ultrasound-specific factors such as probe gain, tissue echogenicity, and scan angle.

To address these challenges and ensure consistent input distributions, we propose an image-specific intensity normalization method as proposed in Loizou et al. [27]. For each image, we manually select two rectangular regions of interest (ROIs) representing low-

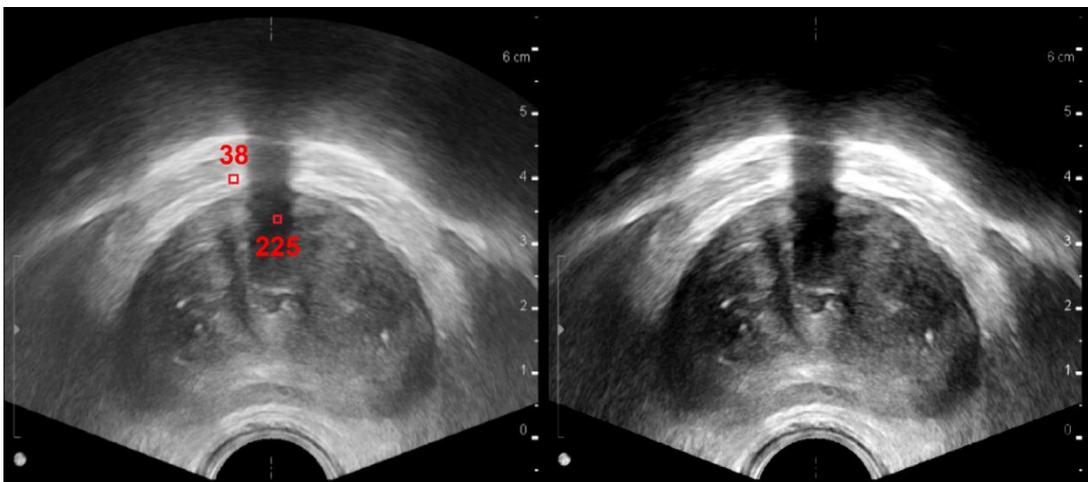


Figure 3-2: Original (O) and normalized (N) ultrasound images of the prostate with manually selected reference regions for intensity normalization [4]

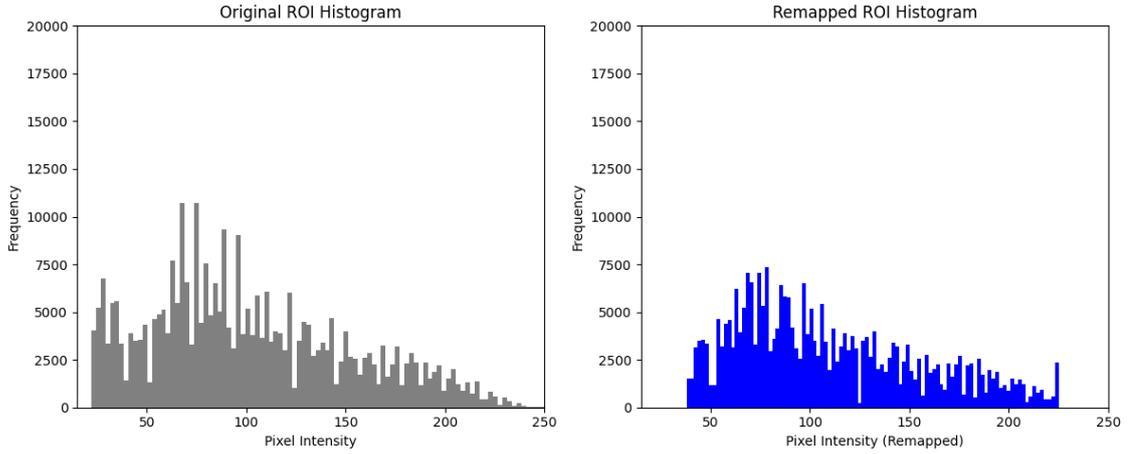


Figure 3-3: Comparison of ROI Pixel Intensity Histograms Before and After Normalization

intensity and high-intensity areas, respectively (see also Fig. 3.2 left). The average intensity within each ROI is computed and used to define the lower and upper bounds of normalization. The pixel values are then normalized according to the following min-max scaling formula:

$$I_{norm} = \frac{I - I_{low}}{I_{high} - I_{low}} \quad (1.1)$$

where I is the original pixel intensity, and I_{low} , I_{high} are the average intensities of the selected low and high intensity ROIs, respectively. This process maps the pixel values of each image to the standardized range of $[0, 1]$, effectively aligning the contrast and brightness distributions across different samples.

Figure 3-2 illustrates the normalization process. In the original TRUS image (left), two representative ROIs are marked in red, with average intensities of 38 and 225, respectively. The original pixel intensity distribution spans from approximately 15 to 250. By applying the above normalization using $I_{low} = 38$ and $I_{high} = 225$, the entire image is remapped to a standardized scale, as shown in the normalized result on the right.

To further assess the effect of normalization within anatomically relevant regions, we compute and compare intensity histograms for a manually delineated region of interest (ROI) before and after normalization. As shown in Figure 3-3, the original intensity distribution within the ROI ranged from 15 to 250, with a highly variable spread. After normalization, the pixel values are redistributed and constrained to the new range of 38

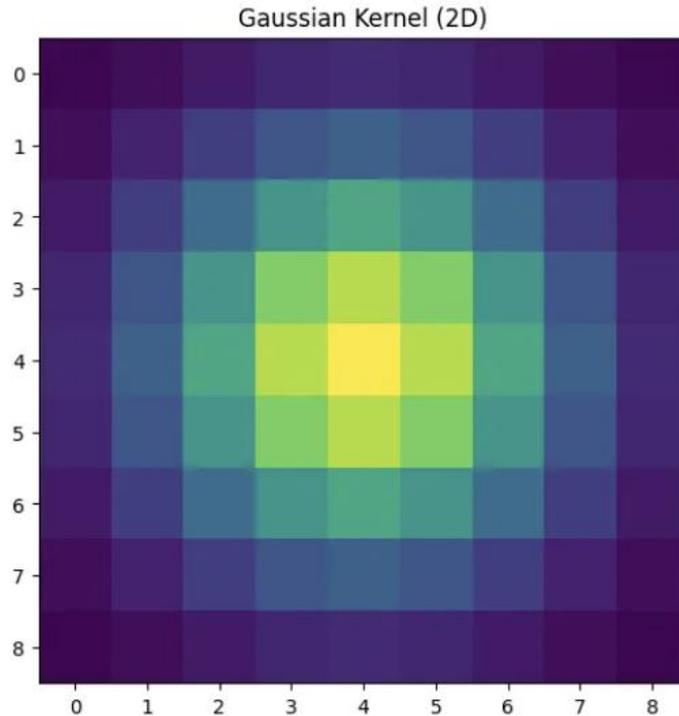


Figure 3-4: Heatmap of a 2D Gaussian Kernel Used for Image Smoothing

to 225, resulting in a more compact and contrast-enhanced distribution. This not only mitigates intensity variability but also facilitates better feature extraction by the segmentation model.

In summary, the proposed ROI-driven intensity normalization method effectively adapts to the intrinsic variability of ultrasound imaging and improves the consistency of intensity profiles across samples. This contributes to a more stable learning process and enhances the network's ability to extract discriminative features for accurate segmentation.

3.2.2 Despeckling

Ultrasound imaging is inherently affected by speckle noise, a granular interference pattern caused by the constructive and destructive interference of backscattered echoes [6]. This type of multiplicative noise not only degrades the visual quality of the image but also adversely impacts the performance of downstream tasks such as semantic segmentation. To mitigate these effects, we incorporate Gaussian filtering [28] during the preprocessing stage as a standard noise reduction strategy.

Gaussian filtering is a classical linear smoothing technique that involves convolving the image with a two-dimensional Gaussian kernel. The kernel assigns weights to

neighboring pixels based on their distance from the center, with closer pixels receiving higher weights. This spatial weighting allows the filter to suppress high-frequency variations (i.e., noise) while preserving edge and structural information in the image [29]. As shown in Figure 3-4, a 9×9 Gaussian kernel exhibits a symmetric bell-shaped distribution, where the center has the highest intensity (weight) and the values gradually decay towards the edges. This enables local smoothing that respects spatial locality, making it well-suited for medical image denoising.

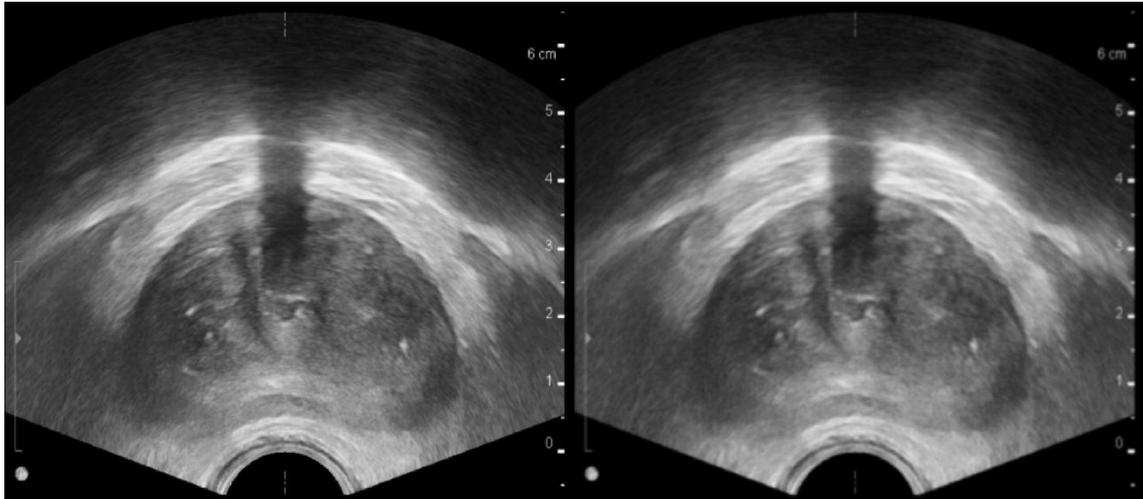


Figure 3-5: Comparison of Original (O) and Despeckled (D) Ultrasound Images

In our implementation, a Gaussian filter with a standard deviation (σ) of 1.2 was applied once to each image. This configuration was empirically selected to achieve a desirable balance between noise suppression and edge preservation. Figure 3-5 presents a visual comparison between the original ultrasound image (left) and the filtered result (right). It is evident that speckle noise has been significantly reduced in the filtered image, while the anatomical structures—particularly the prostate boundaries and internal tissue texture remain well preserved. This highlights the filter’s effectiveness in enhancing image clarity without sacrificing important diagnostic content.

In summary, the use of Gaussian filtering with a moderate kernel size and standard deviation provides an effective means to reduce speckle noise in ultrasound images. It serves as a critical preprocessing step that improves the consistency and quality of the input data for subsequent segmentation tasks.

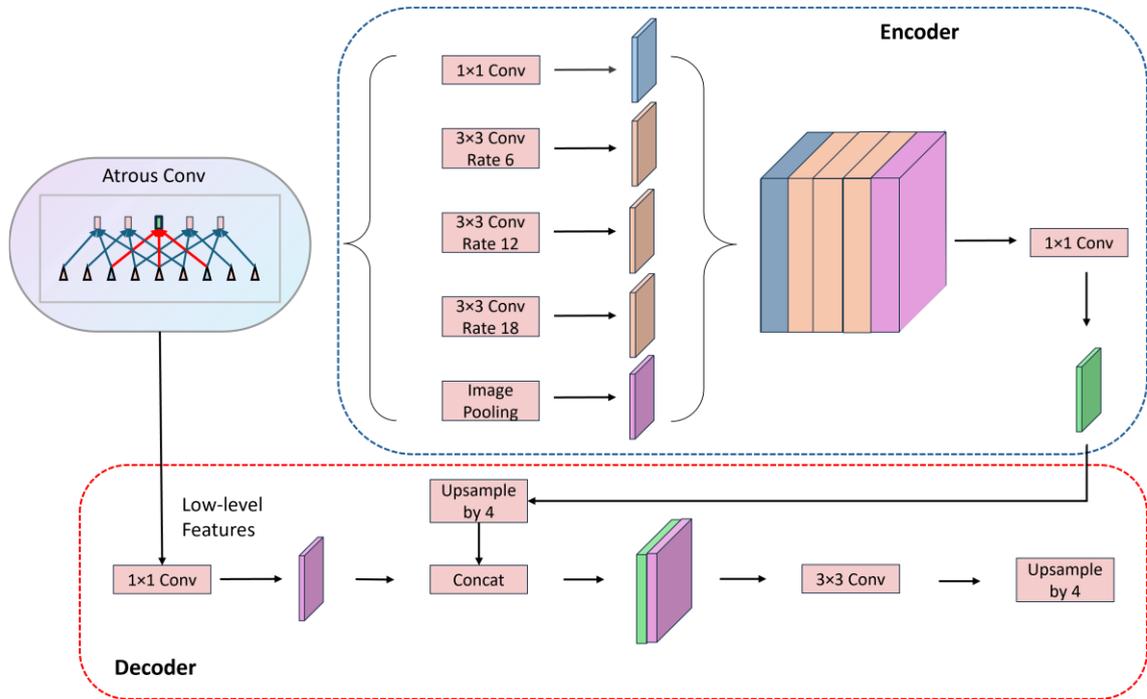


Figure 3-6: Architecture of the Proposed Semantic Segmentation Network [8]

3.3 Segmentation Algorithm

3.3.1 DeepLabv3+ architecture

In this study, we adopt the DeepLabv3+ architecture [19] as the core segmentation network for processing transrectal ultrasound (TRUS) images. DeepLabv3+ is a state-of-the-art deep convolutional neural network designed for semantic segmentation, and it effectively combines the advantages of atrous convolution and encoder–decoder architecture. Its design enables robust multi-scale feature extraction while maintaining accurate localization of object boundaries, which is essential for segmenting complex and ambiguous anatomical structures in ultrasound images [30].

As illustrated in Figure 3-6, the architecture consists of two main components: the encoder and the decoder. The encoder is typically based on a backbone such as ResNet, and incorporates an Atrous Spatial Pyramid Pooling (ASPP) module, which applies atrous convolutions with varying dilation rates (e.g., 6, 12, 18) in parallel, along with a global average pooling branch [19]. These parallel operations allow the model to capture contextual information at multiple scales without sacrificing spatial resolution. The

outputs of ASPP are concatenated and passed through a 1×1 convolution to form the final high-level feature representation.

The decoder module is introduced in DeepLabv3+ to refine segmentation details, particularly along object boundaries. It first extracts low-level features from earlier layers of the backbone, which retain fine spatial details. These features are passed through a 1×1 convolution and then concatenated with the upsampled high-level features from the encoder. This fused representation is then processed through a 3×3 convolution to enhance spatial coherence, followed by another upsampling step to produce the final prediction at the original image resolution.

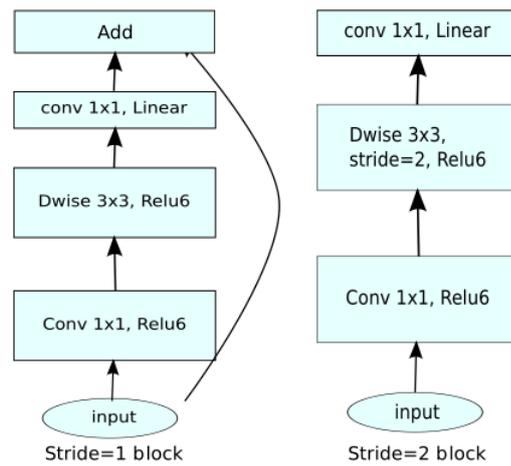


Figure 3-7: Structure of the Basic Building Blocks in MobileNetV2 [10]

We selected DeepLabv3+ for its strong ability to handle noisy textures, fuzzy boundaries, and variable-scale features, all of which are common challenges in TRUS image segmentation. The architecture’s ability to combine deep semantic understanding with fine-grained spatial detail makes it particularly well suited for accurate prostate delineation in ultrasound images.

In the original implementation of DeepLabv3+, the authors employed the **Xception** network [31] as the backbone to extract deep features due to its strong representational capacity [32].

To improve efficiency while maintaining satisfactory segmentation performance, we replace Xception with **MobileNetV2** [33], a lightweight convolutional neural network specifically designed for speed and low memory footprint. MobileNetV2 is based on **inverted residual blocks with linear bottlenecks**, which consist of a sequence of

Table 3-1: Modified MobileNetV2 Architecture Used as the Backbone in the Proposed Network [12]

Input Size	Operator	t	c	n	s
$224^2 \times 3$	conv2d 3×3	–	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck (dilated)	6	96	3	1
$14^2 \times 96$	bottleneck (dilated)	6	160	3	1
$14^2 \times 160$	bottleneck (dilated)	6	320	1	1
$14^2 \times 320$	conv2d 1×1	–	1280	1	1

convolutions: an initial 1×1 expansion layer with ReLU6 activation, a depthwise separable 3×3 convolution, and a final 1×1 linear projection. When the stride is 1 and the input/output channels match, a residual connection is added. This structure is illustrated in Figure 3-7.

In our implementation as proposed in this work, we adapt MobileNetV2 to the segmentation task by removing the classification head, exposing intermediate low-level and high-level features, and replacing late-stage downsampling with dilated convolutions to preserve spatial resolution (with a downsampling factor of 16). The modified network architecture is detailed in Table 3-1.

By employing MobileNetV2 as the encoder, we achieve a significant reduction in model complexity and computational overhead, making the overall network more efficient for high-resolution ultrasound segmentation tasks. At the same time, the use of dilated convolutions and a DeepLabv3+-style decoder ensures sufficient receptive field and accurate boundary delineation [34].

3.3.2 Muti-Input

3.3.2.1 Binarized and initial contour images

To generate the initial boundary cues for our multi-input framework, we perform a three-step preprocessing pipeline as illustrated in Figure 3-9: Channel Expansion and Fusion Prior to Backbone Input. First, we apply intensity normalization to the original ultrasound



Figure 3-8: Illustration of Pre-processing Steps: Normalization (N), Binarization (B), and Initial Contour Extraction (C) in an ultrasound prostate image

images to reduce inter-sample intensity variability and enhance the structural contrast. The N image is shown on the left of the figure.

Next, based on the N image, a manually selected global threshold is applied to convert the image into a binary mask (middle image). This threshold is empirically determined to capture the most prominent echogenic regions across the dataset and is then fixed for batch-level binarization.

Finally, we extract the structural boundaries from the binary mask using contour detection, a standard image processing technique implemented via `cv2.findContours` in OpenCV [35]. The resulting contour map (right image) highlights the initial structural outlines in green, which serve as an auxiliary spatial input in our subsequent multi-stream learning architecture.

3.3.2.2 *Channel Expansion*

As illustrated in Figure 3-9, we introduce a channel expansion strategy to incorporate structural priors into the input of our network. In this design, a binary structural cue—such as an initial contour map extracted from the normalized ultrasound image—is concatenated with the original 3-channel ultrasound image along the channel axis (i.e., $\text{dim}=1$), forming a 4-channel composite input [36]. To ensure compatibility with conventional backbone networks (e.g., MobileNetV2 or Xception), which are typically designed to accept 3-channel inputs, a 1×1 convolutional layer is applied to compress the 4-channel input back to 3 channels. This approach preserves both the original image intensity and the spatial contour information, enabling early fusion of complementary modalities. By introducing this structural guidance at the input level, the network is able

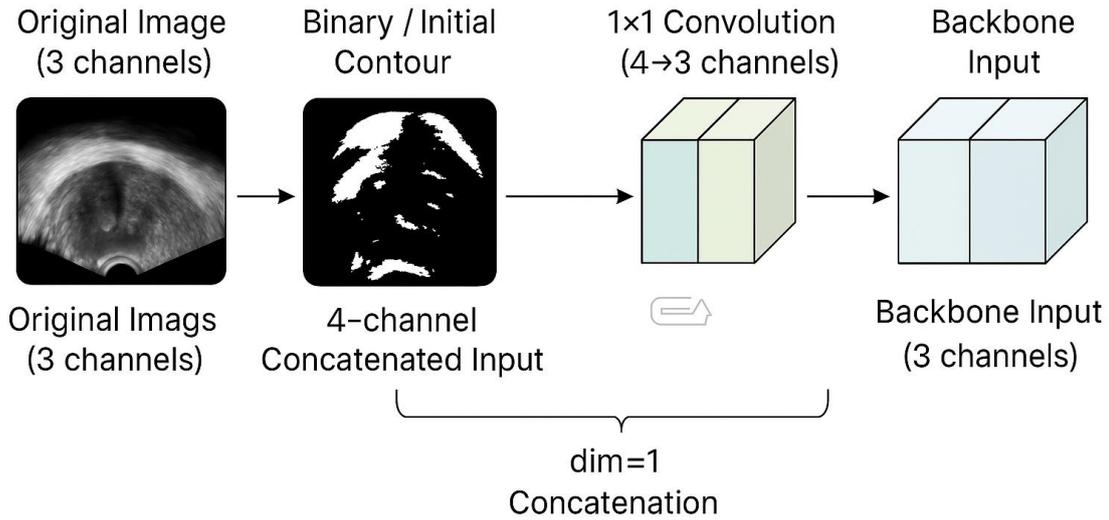


Figure 3-9: Channel Expansion and Fusion Prior to Backbone Input [14]

to more effectively focus on relevant anatomical regions in subsequent stages of feature extraction.

3.3.2.3 Dual-Branch

To enable comparative experiments with different input combinations, we adopt a generalized dual-input architecture where the two inputs are denoted as Input_1 and Input_2, as illustrated in Figure 3-10. Input_1 is fed into the main encoder path, which is composed of a DCNN backbone equipped with Atrous Spatial Pyramid Pooling (ASPP) to extract rich multi-scale semantic features. Meanwhile, Input_2 is passed through a Lightweight Context Module that captures both mid-range and global contextual information using dilated convolution and global average pooling [37]. The processed features from Input_2 are further refined through a 1×1 convolution and then used to enhance the shallow features from the encoder via feature fusion. This flexible design supports the integration of various structural or modality cues (e.g., grayscale, binary, edge maps) and facilitates a fair evaluation of how different auxiliary inputs influence segmentation performance.

3.3.3 Automatic Hyperparameter Optimization Algorithm

To efficiently identify optimal hyperparameters, we adopt a two-stage auto-tuning strategy that combines grid search and Bayesian optimization [38].

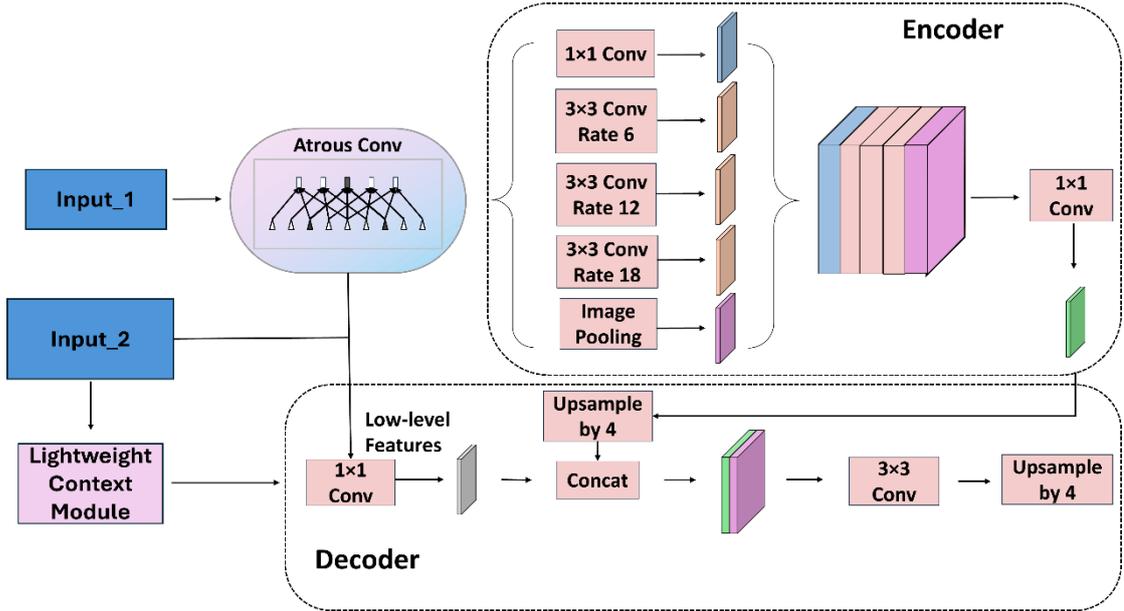


Figure 3-10: Overall Architecture of the Proposed Network with Dual Input and Contextual Feature Fusion [ref]

In the first stage, we employ grid search, a brute-force method [39] that exhaustively evaluates all predefined combinations of discrete hyperparameters across a fixed grid. While simple and effective for low-dimensional, structured search spaces, grid search is computationally expensive and inefficient in the presence of unimportant or redundant parameters (see Figure 3-10, left). In our implementation, grid search is applied to explore discrete parameters such as batch size ($\{8, 16\}$), optimizer (Adam, SGD), and a coarse range of learning rates ($1e-4$ to $7e-3$) [40].

Once the best-performing combination from the discrete space is identified, the second stage uses Bayesian optimization to refine continuous parameters—specifically, the learning rate. Bayesian optimization constructs a surrogate model, often using Gaussian Processes, to approximate the objective function. An acquisition function, specifically the Expected Improvement (EI) function (1.2) is then used to select the most promising hyperparameter configurations for evaluation, balancing exploration and exploitation. As shown in Figure 3-11 (right), this strategy allows the search to focus adaptively on regions likely to contain optima, significantly reducing the number of evaluations needed.

$$EI(x) = (f(x^+) - \mu(x))\Phi\left(\frac{f(x^+) - \mu(x)}{\sigma(x)}\right) + \sigma(x)\phi\left(\frac{f(x^+) - \mu(x)}{\sigma(x)}\right) \quad (1.2)$$

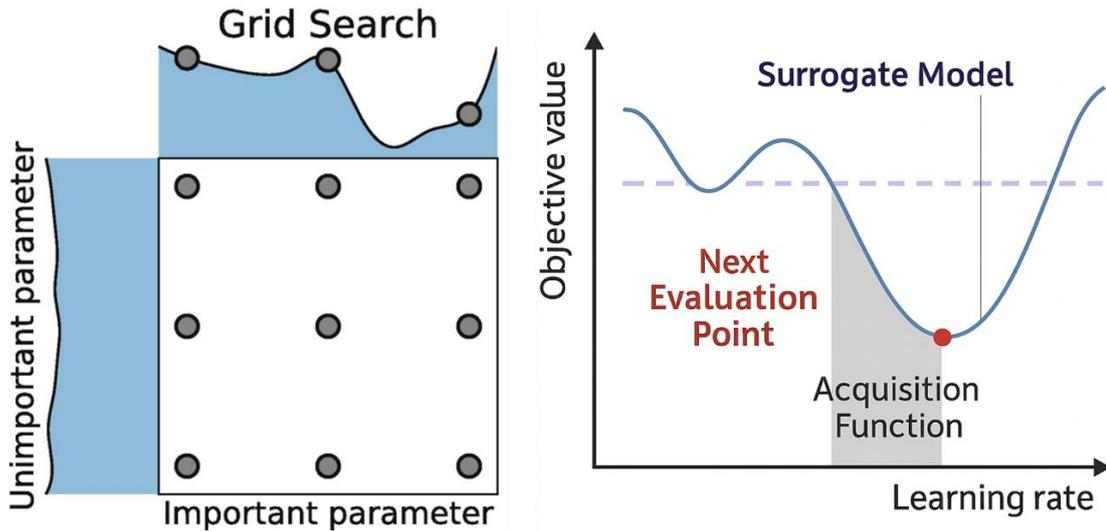


Figure 3-11: Illustration of the Automatic Hyperparameter Optimization Method [19], [20]

$\mu(x)$ and $\sigma(x)$ are the predicted mean and standard deviation of the surrogate model at point x . $\Phi(\cdot)$ is the cumulative distribution function (CDF) of the standard normal distribution, $\phi(\cdot)$ is the probability density function (PDF) of the standard normal distribution, and $f(x^+)$ denotes the best objective value observed so far [41], [42].

By combining grid search for robust discrete-space exploration and Bayesian optimization for sample-efficient continuous-space refinement, our method achieves strong parameter tuning performance while maintaining computational efficiency.

3.3.4 Early stop

To prevent overfitting and improve training efficiency, we employ an early stopping strategy during model training. Early stopping is a widely used regularization technique that monitors the validation performance (e.g., mIoU) over epochs and halts training when no further improvement is observed. The core idea is to stop the optimization process once the model begins to overfit the training data, thereby preserving the generalization capability of the network [43].

In our implementation, we monitor the validation mIoU and define a stopping criterion based on three parameters: the patience, the minimum delta, and the warm-up period. Specifically, we set a warm-up period of 80 epochs during which early stopping is

disabled to ensure sufficient initial training. After this phase, the training loop checks if the validation score has improved by at least a minimum delta of 0.0005. If no significant improvement is observed for 20 consecutive epochs (i.e., patience), training is terminated automatically. The best model is determined by the highest recorded mIoU during the training process. This method allows us to reduce unnecessary computation while maintaining robust model performance.

3.3.5 Evaluation Metrics for the Segmentation

To quantitatively evaluate the performance of the proposed deep learning (DL) segmentation method, we computed six standard evaluation metrics (EMs) to compare the agreement between the automated segmentation results and the manually annotated ground truth. These metrics include the Dice Coefficient (DC), which measures spatial overlap and is particularly suitable for imbalanced segmentation tasks; Precision (P), reflecting the proportion of correctly predicted positive pixels among all predicted positives; Specificity (S), indicating the model's ability to correctly identify negative regions; and Accuracy (AC), which measures the overall proportion of correctly classified pixels [44]. Additionally, we calculated Cohen's Kappa coefficient (CK) to assess the agreement beyond chance between predicted and ground truth labels, and the Hausdorff Distance (HD), which quantifies the maximum surface distance between the predicted and reference contours and is useful for detecting boundary-level discrepancies [45].

The overall effectiveness of the segmentation model was assessed by averaging these six metrics across all test images within each fold of a three-fold cross-validation scheme, and for each of the four image preprocessing configurations. This ensured a comprehensive and fair comparison of the model's performance under varying input conditions.

To further evaluate differences between the segmentation outcomes obtained under different preprocessing strategies, we applied additional statistical analyses. Specifically, the Wilcoxon rank-sum test (non-parametric, $p < 0.05$) was used to assess the statistical significance of segmentation performance differences between image groups. The Spearman rank correlation coefficient (ρ) was also computed to investigate monotonic relationships between model predictions and manual annotations [46].

Prior to inferential analysis, we assessed the distributional properties of all variables using two widely adopted normality tests: the Kolmogorov–Smirnov test [47] and the Shapiro–Wilk test [48]. As most distributions significantly deviated from normality ($p < 0.05$), we reported all quantitative results in terms of median \pm interquartile range (IQR) instead of mean \pm standard deviation, to better reflect central tendency and dispersion in non-Gaussian data.

In addition to these tests, box plots were utilized for visual inspection of the metric distributions across different groups, clearly illustrating variability, outliers, and overall data spread. We also performed Bland–Altman analysis [13], a standard method for evaluating agreement between two measurement methods, to examine the consistency between the manual and automated segmentations, as well as among different preprocessing strategies. Furthermore, a linear regression analysis was conducted to explore potential trends and predictive relationships between the different methods, providing deeper insight into the behavior of the segmentation models across varying preprocessing pipelines.

4 Results

In this chapter the results of the proposed segmentation algorithm are presented using tables and bar charts in order to illustrate the applicability of the algorithm. The settings of the algorithm are given in subsection 3.3. Figure 4-1 presents a visual comparison of the segmentation performance of the proposed DL method (see subsection 3.3) under four different image preprocessing strategies: O, N, D, and ND. Each column corresponds to one preprocessing method, with the top row showing examples with clear prostate boundaries and the bottom row showing challenging cases with blurry boundaries. In each image, the white dashed line represents the automatic segmentation result from the model, while the white solid line denotes the manually annotated ground truth.

Quantitative metrics including the DC, AC are reported within each image. The N strategy yielded the best results in clear-boundary cases, achieving a DSC of 98.1% and an AC of 99.2%. For the blurry-boundary cases, the ND strategy showed superior robustness, with a DSC of 85.3% and an AC of 95.6%. In contrast, the model's performance on O images in blurry scenarios dropped to a DSC of 82.0% and an AC of 94.3%.

Figure 4-2 presents box plots illustrating the distribution of six EM used to assess the performance of the proposed DP segmentation method on the N PCa ultrasound images. The N PCa images were selected as were those showed best performance compared with the rest of the preprocessing schemes (O, D, ND). The horizontal axis represents the different metrics, including the DC, P, S, AC, CK, and HD. Each box plot depicts the statistical distribution of the corresponding metric across all test images, with the median and IQR range annotated as (median, IQR) below each box plot.

The detailed median±IQR results for each EM are as follows: DC = 0.949 ± 0.038 , P = 0.964 ± 0.058 , S = 0.994 ± 0.009 , AC = 0.984 ± 0.008 , CK = 0.938 ± 0.042 , and HD = 3.378 ± 1.739 mm. The left vertical axis corresponds to the normalized metrics (DC, P, S, AC, CK), ranging from 0.8 to 1.0, while the right vertical axis indicates the HD values in millimeters. As shown in the boxplot, the interquartile ranges (IQR) of the Specificity (S) and Accuracy (AC) metrics are notably smaller compared to those of other metrics such as Dice coefficient (DC) and Cohen's Kappa (CK). This indicates that S and AC have higher stability and less variability across the test set, suggesting consistently high performance in these metrics. On the contrary, Precision (P) and CK exhibit relatively

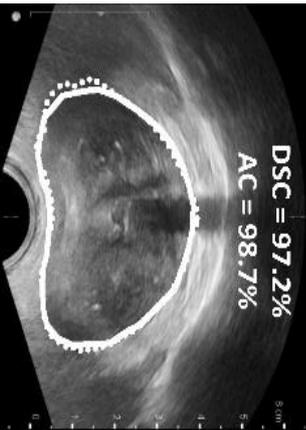
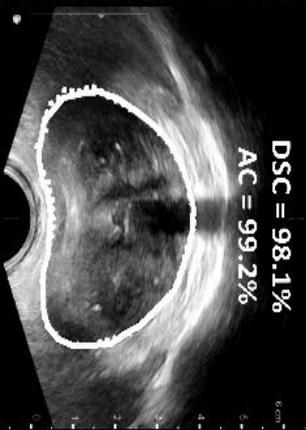
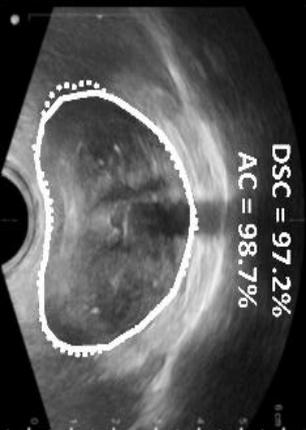
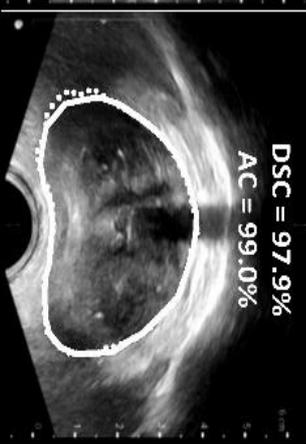
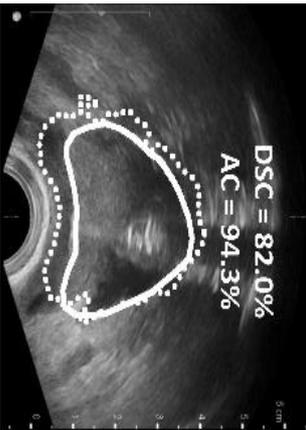
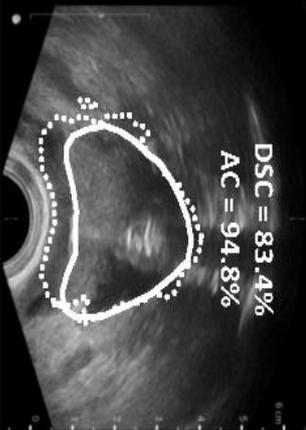
O	N	D	ND
 <p>DSC = 97.2% AC = 98.7%</p>	 <p>DSC = 98.1% AC = 99.2%</p>	 <p>DSC = 97.2% AC = 98.7%</p>	 <p>DSC = 97.9% AC = 99.0%</p>
 <p>DSC = 82.0% AC = 94.3%</p>	 <p>DSC = 88.4% AC = 96.8%</p>	 <p>DSC = 83.4% AC = 94.8%</p>	 <p>DSC = 85.3% AC = 95.6%</p>

Figure 4-1: Segmentation performance comparison across different preprocessing strategies on clear (top row) and ambiguous (bottom row) prostate boundaries

larger IQRs, reflecting greater variability in segmentation outcomes.

Regarding the median positions, the medians for most metrics, including DC, S, AC, and CK, are approximately centered within their respective boxes, indicating a relatively symmetric distribution. However, for Precision (P), the median appears slightly skewed toward the upper boundary of the box, suggesting a slight negative skewness in the distribution.

Overall, the smaller IQRs and centered medians in S and AC metrics demonstrate the robustness of the segmentation model, while the slightly larger spread in other metrics highlights the areas where performance consistency can still be further improved.

Table 4-1 presents a comprehensive statistical summary of the segmentation performance of the proposed DL method across four preprocessing schemes (O, N, D, NDo) on a test set consisting of 58 TRUS images. The EM evaluated include DC, P, S, AC, CK, and the HD, each reflecting different aspects of segmentation accuracy and consistency.

Following the presentation of Table 4-1, which summarizes the segmentation performance across different preprocessing schemes, several observations can be made. Firstly, all evaluated metrics (DC, P, S, AC, CK) exhibited high median values above 92%, indicating that the proposed DL method consistently achieved good segmentation performance across the dataset. Among them, specificity (S) achieved the highest median value (99.4%), suggesting excellent ability to correctly identify non-boundary regions.

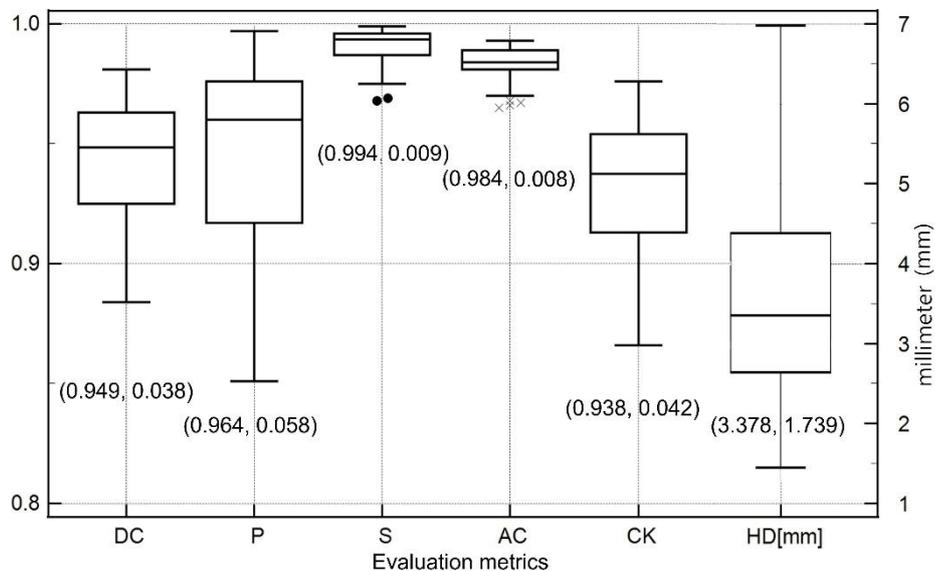


Figure 4-2: Box plots of segmentation EM for N PCa images

In terms of variability, the interquartile range (IQR) values show that metrics such as S (IQR = 1.35%) and AC (IQR = 0.98%) had very narrow spreads, implying high consistency of model performance among different test images. In contrast, precision (P) displayed a relatively larger IQR (8.36%), indicating greater variation in the model's ability to accurately predict boundary pixels under different conditions.

The Hausdorff Distance (HD), which is measured in millimeters, had a median value of 3.083 mm with a comparatively larger IQR of 1.881 mm, reflecting some variability in boundary localization accuracy, especially in more challenging cases.

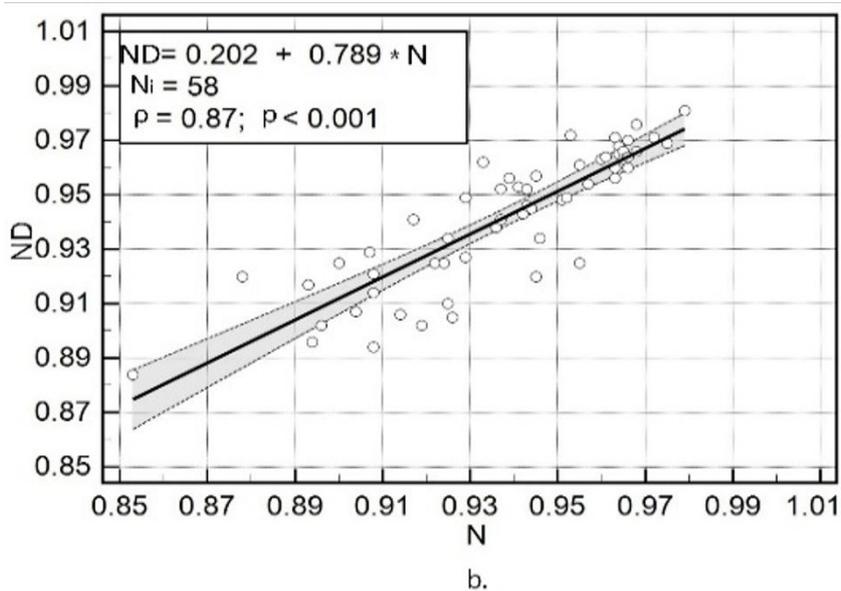
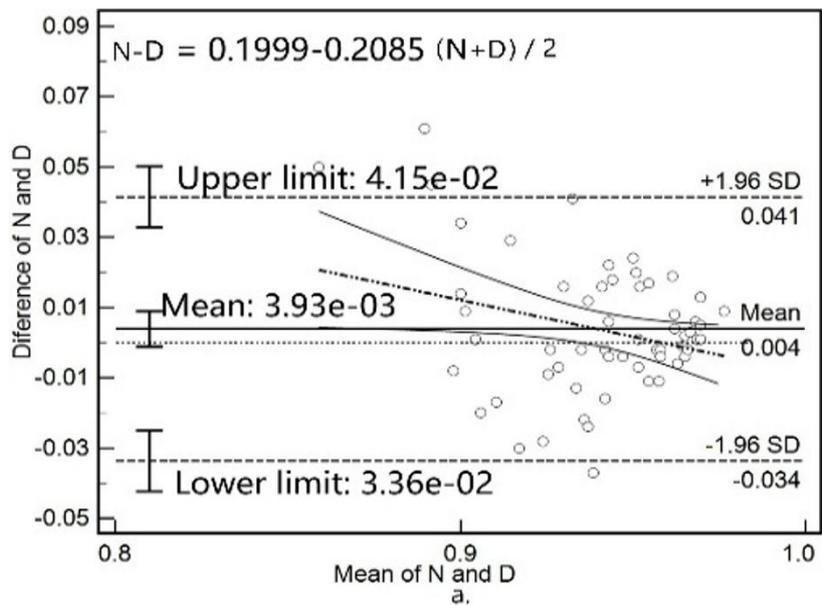


Figure 4-3: Bland-Altman plot in Fig. 4-3a) and linear regression analysis in Fig. 4-3b), of the DC under different preprocessing methods (N vs D)

Table 4-1: Statistical summary of segmentation metrics across four preprocessing methods on the whole test set of PCa images investigated in this study (N=58)

EM	DC [%]	P [%]	S [%]	AC [%]	CK [%]	HD [mm]
	O/N/D/ND	O/N/D/ND	O/N/D/ND	O/N/D/ND	O/N/D/ND	O/N/D/ND
Median	94.02/94.84/94.43/94.	93.50/96.00/94.35/9	98.87/99.34/98.90	98.24/98.40/98.35	92.89/93.77/93.16	3.68/3.36/3.51/3.
	22	5.56	/99.14	/98.36	/92.92	76
IQR	3.93/3.92/3.05/4.19	8.35/6.01/6.50/6.48	1.35/0.95/1.35/1.0	0.98/0.77/0.93/0.8	3.71/4.21/3.37/4.2	1.88/1.79/1.90/1.
			8	6	2	62
MAX	97.49/98.14/97.19/97.	99.69/99.75/99.02/9	99.95/99.93/99.73	99.26/99.33/99.22	96.77/97.59/96.35	7.94/6.98/8.73/7.
	93	9.84	/99.96	/99.32	/97.31	29
MIN	81.98/88.41/83.43/85.	70.34/85.14/72.61/7	93.67/96.80/94.35	94.27/96.51/94.84	78.69/86.56/80.45	1.64/1.45/2.05/1.
	25	6.64	/95.52	/95.59	/82.70	33
Kur.	242.11/-	331.83/-	658.60/169.88/54	477.34/101.22/56	308.47/-	0.47/0.06/1.60/0.
	77.80/228.10/49.85	41.07/290.16/243.21	3.19/267.22	6.40/199.60	54.54/305.49/99.3	26
			5			
Skew.	-138.68/-52.58/-	-155.08/-79.38/-	-220.21/-131.46/-	-190.45/-114.93/-	-149.52/-52.72/-	0.90/0.68/1.18/0.
	146.95/-82.48	164.85/-149.79	182.25/-147.76	180.80/-110.26	154.79/-86.63	57

EM: Evaluation metrics, DC: Dice similarity coefficient, P: Precision, S: Specificity, AC: Accuracy, CK: Cohen's Kappa, HD: Harsdorf Distance, Max, Min: Maximum and minimum values, Kur.: Kurtosis, Skew.: Skewness

Skewness and kurtosis values provided additional insights into distribution characteristics: metrics like DC and CK showed relatively symmetrical distributions, whereas HD presented a higher skewness and kurtosis, indicating a few extreme outliers where the segmentation boundary differed more significantly from the ground truth.

Overall, Table 4-1 highlights that while the proposed method performs robustly overall, certain metrics and extreme cases warrant further attention to optimize boundary delineation accuracy in difficult ultrasound images.

Figure 4-3 presents a comprehensive analysis of the consistency and correlation of the DC across different preprocessing methods for prostate ultrasound image segmentation using the proposed DL approach. The figure is divided into two subplots, where Figure 4-3(a) shows a Bland-Altman analysis comparing the segmentation results obtained from two preprocessing methods (N vs D). The Bland Altman method is commonly used to assess the agreement between two measurement techniques. The x-axis represents the mean DC values between the N and D methods for each image, while the y-axis shows the difference in DC values (N - D). The plot demonstrates that most data points lie within the limits of agreement, with a mean difference of 0.0039, an upper limit of 0.0415, and a lower limit of -0.0336. These results indicate that the N and D preprocessing methods yield highly consistent segmentation performance, with only minor deviations across samples. The equation shown in the top-left corner of Figure 4.3(a),

$ND = 0.1999 - 0.2085 \cdot \frac{(N + D)}{2}$, represents a regression model that approximates the relationship between the difference in DC values (N - D) and their mean. This negative slope suggests that as the average segmentation performance of the two methods increases, the difference between them slightly decreases. In other words, the agreement between N and D becomes more stable when segmentation accuracy is high, further supporting the consistency of these preprocessing approaches.

Figure 4-3(b) illustrates a linear regression analysis between the DC values obtained from the N and ND preprocessing methods. The x-axis represents the DC from the N method, and the y-axis represents the DC from the ND method. The regression line follows the

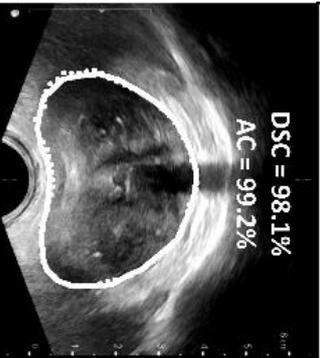
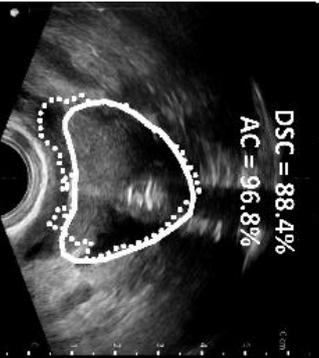
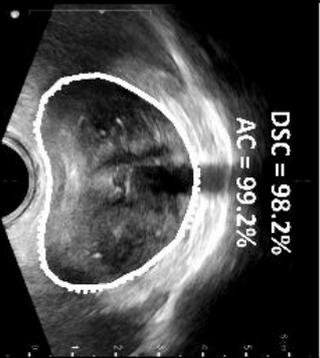
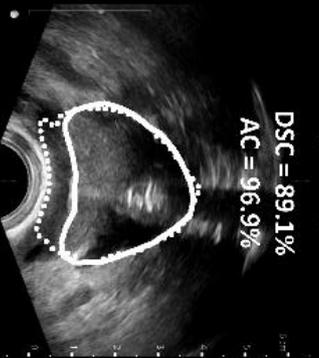
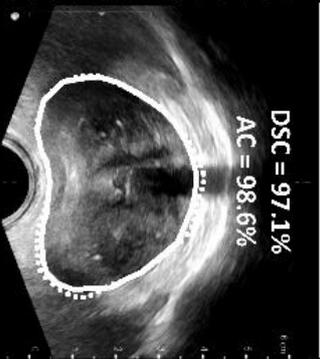
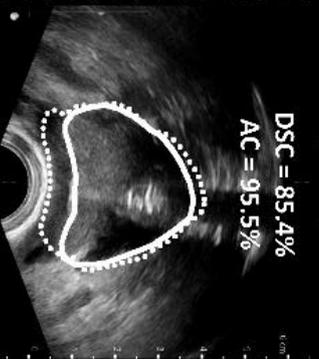
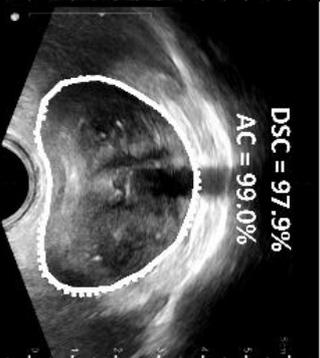
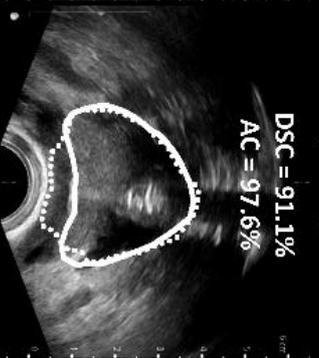
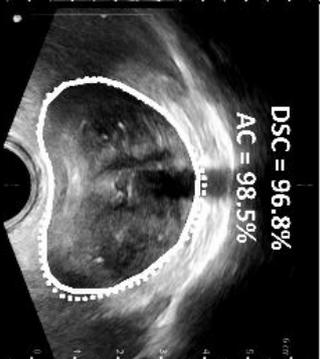
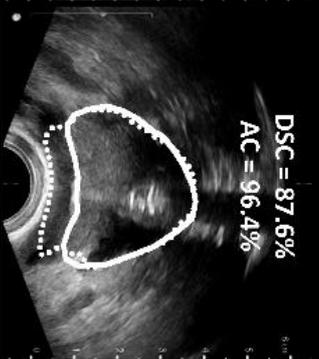
N	NBD	NBE	NCD	NCE
 <p>DSC = 98.1% AC = 99.2%</p>  <p>DSC = 88.4% AC = 96.8%</p>	 <p>DSC = 98.2% AC = 99.2%</p>  <p>DSC = 89.1% AC = 96.9%</p>	 <p>DSC = 97.1% AC = 98.6%</p>  <p>DSC = 85.4% AC = 95.5%</p>	 <p>DSC = 97.9% AC = 99.0%</p>  <p>DSC = 91.1% AC = 97.6%</p>	 <p>DSC = 96.8% AC = 98.5%</p>  <p>DSC = 87.6% AC = 96.4%</p>

Figure 4-4 Enhanced segmentation results under different preprocessing and input fusion strategies following automated hyperparameter optimization.

equation $ND = 0.202 + 0.789 \times N$, indicating a strong linear relationship between the two sets of results. The correlation coefficient is $\rho = 0.87$, with a statistical significance of $p < 0.001$, suggesting a robust positive correlation between the segmentation results from these two preprocessing strategies. The shaded region around the regression line represents the 95% confidence interval, providing visual evidence of the reliability of this relationship.

Figure 4-4 presents a comparative visualization of the segmentation performance under five different preprocessing schemes and input fusion strategies for TRUS prostate images. The five configurations include: N, NBD (normalized image + binary map, dual-branch), NBE (normalized + binary image, channel expansion), NCD (normalized + initial contour map, dual-branch), and NCE (normalized + initial contour map, channel expansion). In each case, segmentation results are shown for two representative images—one with a clear prostate boundary (top row) and one with an ambiguous boundary (bottom row). In all images presented in Fig. 4-4, the white solid line indicates the ground truth (manual segmentation), while the dashed line represents the automatic segmentation predicted by the network. The DSC and AC are annotated on each example to quantitatively reflect the performance.

From the top row in Figure 4-4, we observe that all five methods perform similarly well on images with clear boundaries, achieving high DSC and AC values (e.g., N: 98.1% DSC / 99.2% AC; NBD: 98.2% / 99.2%). However, differences become more evident in the bottom row, where the prostate boundaries are less distinct. Here, the NCD method achieves the highest accuracy (91.1% DSC / 97.6% AC), suggesting its superior ability to incorporate structural guidance from initial contours. Comparatively, the NBE and NCE approaches show slightly reduced performance, highlighting the benefit of using dedicated dual-branch structures over direct channel expansion in complex boundary conditions. Best results were obtained for the NBD images where a DC=0.969 was obtained.

Figure 4-5 illustrates the distribution of the DC across five different preprocessing and input fusion strategies for prostate ultrasound image segmentation. The groups include: N (segmentation with normalized images only), NBD (dual-branch input using normaliz-

Table 4-2: Statistical summary of segmentation metrics comparing normalized input (N) with different dual-input strategies (NBD, NBE, NCD, NCE) on the whole test set of PCa images investigated in this study (N = 58)

EM	DC [%]	P [%]	S [%]	AC [%]	CK [%]	HD [mm]
	N/NBD/NBE/NCD/NCE	N/NBD/NBE/NCD/NC	N/NBD/NBE/NCD/NC	N/NBD/NBE/NCD/NC	N/NBD/NBE/NCD/NC	N/NBD/NBE/NCD/NCE
		E	E	E	CE	
Median	94.84/96.89/96.26/96.77/95.5	96.00/97.07/94.00/96.6	99.34/99.43/98.89/99.47/	98.4/99.09/98.80/99.0	93.77/96.22/95.48/96.	3.36/1.94/2.21/2.00/2.51
	4	8/94.11	98.83	3/98.59	06/94.60	
IQR	3.92/2.38/2.67/2.42/2.62	6.01/4.07/4.99/4.44/5.2	0.95/0.72/1.03/0.80/0.95	0.77/0.52/0.80/0.72/0.	4.21/2.54/2.98/2.71/2.	1.79/1.31/1.43/1.06/1.47
		1		92	80	
MAX	98.14/98.49/98.48/98.79/98.2	99.75/99.59/98.01/99.5	99.93/99.91/94.86/99.94/	99.33/99.61/99.45/99.	97.59/98.14/98.12/98.	6.98/7.07/7.03/7.02/7.02
	3	8/98.94	99.84/	58/99.54/	45/97.81	
MIN	88.41/89.15/85.41/89.27/82.0	85.14/82.41/74.76/82.0	96.81/96.83/94.86/97.00/	96.51/96.87/95.49/97.	86.56/87.33/82.81/87.	1.45/0.68/1.14/0.60/0.94
	3/87.64	3/80.70	95.74	12/96.22	62/85.55	
Kur.	-	-0.41/5.11/5.03/1.00	1.70/4.48/3.49/1.30/2.11	1.01/3.75/5.06/1.82/1.	-	0.06/5.55/3.00/4.14/1.14
	0.78/3.88/5.66/1.65/3.19/1.36				88	0.55/4.64/6.33/1.95/1.
					78	
Skew.	-0.53/-1.59/-2.12/-1.3/-1.69/-	-0.79/-1.94/-1.99/-1.14	-1.31/-1.87/-1.72/-1.27/-	-1.15/-1.62/-2.03/-	-0.53/-1.71/-2.23/-	0.68/1.98/1.63/1.86/1.18
	1.18		1.39	1.25/-1.33	1.33/-1.26	

EM: Evaluation metrics, DC: Dice similarity coefficient, P: Precision, S: Specificity, AC: Accuracy, CK: Cohen's Kappa, HD: Hausdorff Distance, Max: Min: Maximum and minimum values, Kur.: Kurtosis, Skew.: Skewness

ed and binary images), NBE (channel expansion with normalized and binary images), NCD (dual-branch input using normalized and initial contour images derived from the binary map), and NCE (channel expansion with normalized and initial contour images). Each box plot represents the DC distribution for the corresponding strategy across all test images. The values in parentheses below each plot indicate the median and interquartile range (\pm IQR) for that group. Outliers are marked individually, providing insight into the stability of segmentation performance under each method. This visualization facilitates a clear comparison of central tendency and variability among the evaluated methods.

Figure 4-6 contains two subplots illustrating the consistency and correlation of DC values obtained from the N and NBD groups. Figure 4-6(a) on the left displays a Bland-Altman plot, where the x-axis represents the mean DC values of the two methods

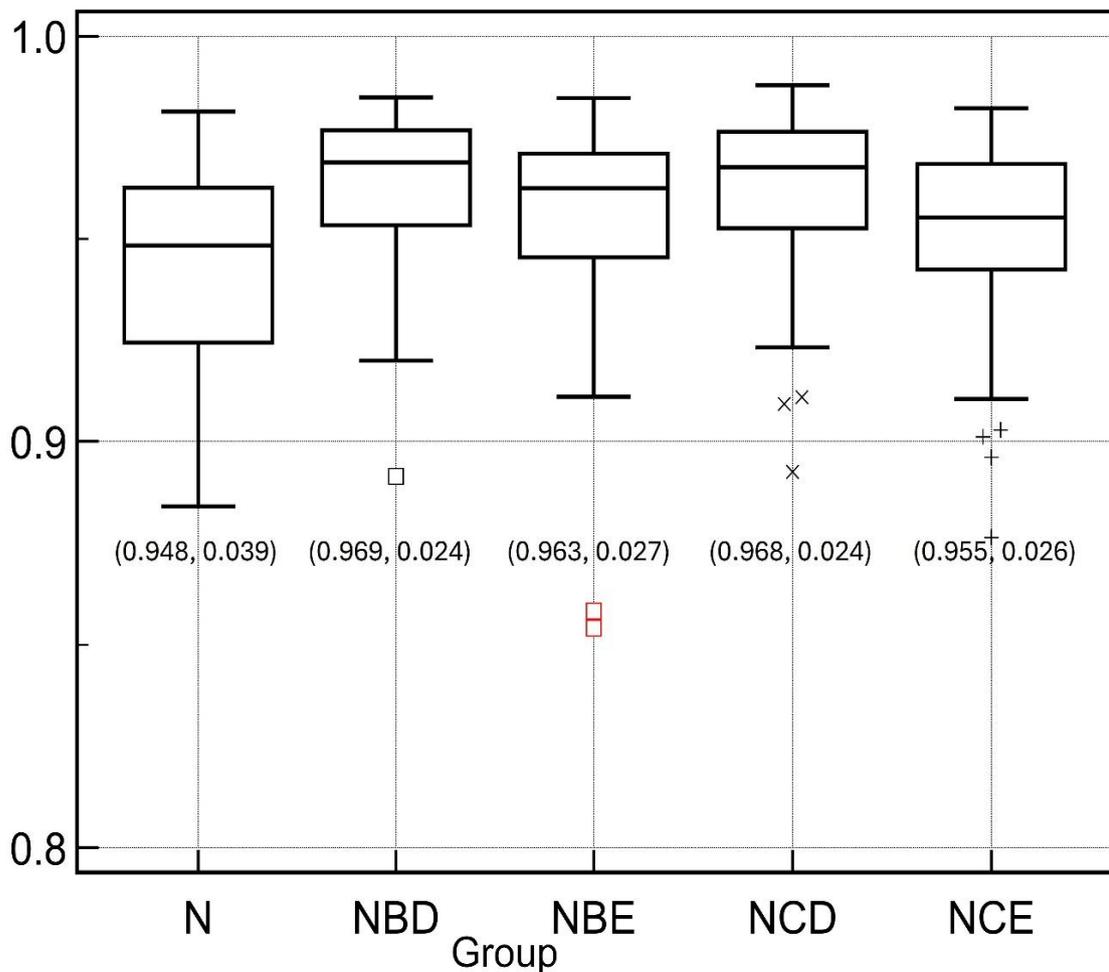


Figure 4-5: Box plots of the DC for different input fusion strategies on images investigated in this study (N=58). The (median, IQR) are shown below each boxplot.

and the y-axis shows the difference (NBD-N). The central solid line represents the mean difference, while the upper and lower dashed lines indicate the limits of agreement (± 1.96

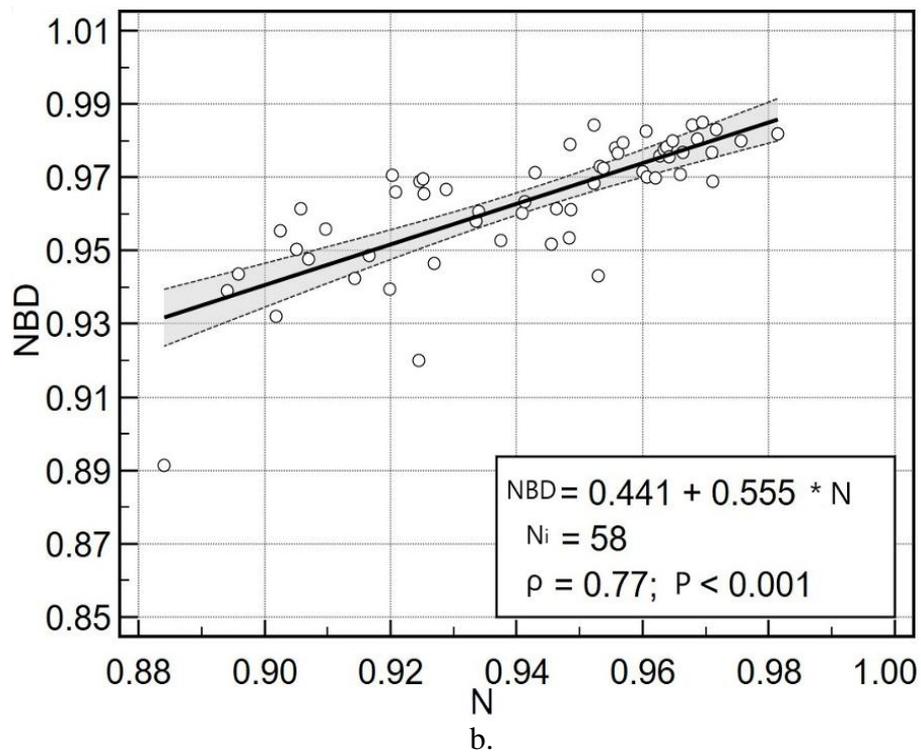
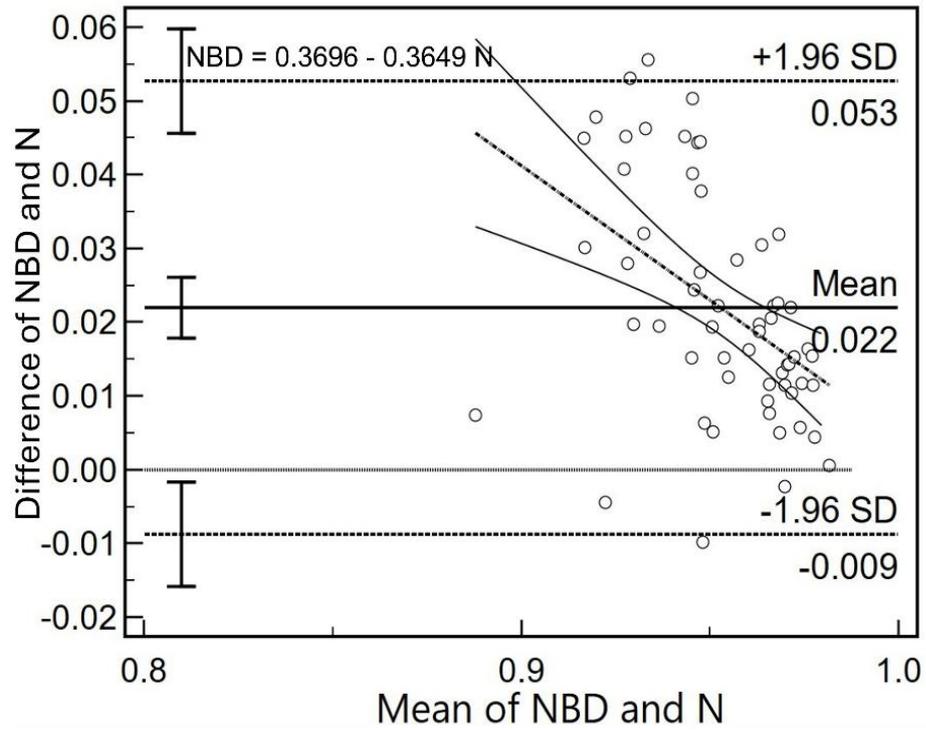


Figure 4-6: Bland-Altman plot (see Fig. 4-6a) and correlation analysis of the DC (see Fig. 4-6b) between the N and NBD groups on images investigated in this study.

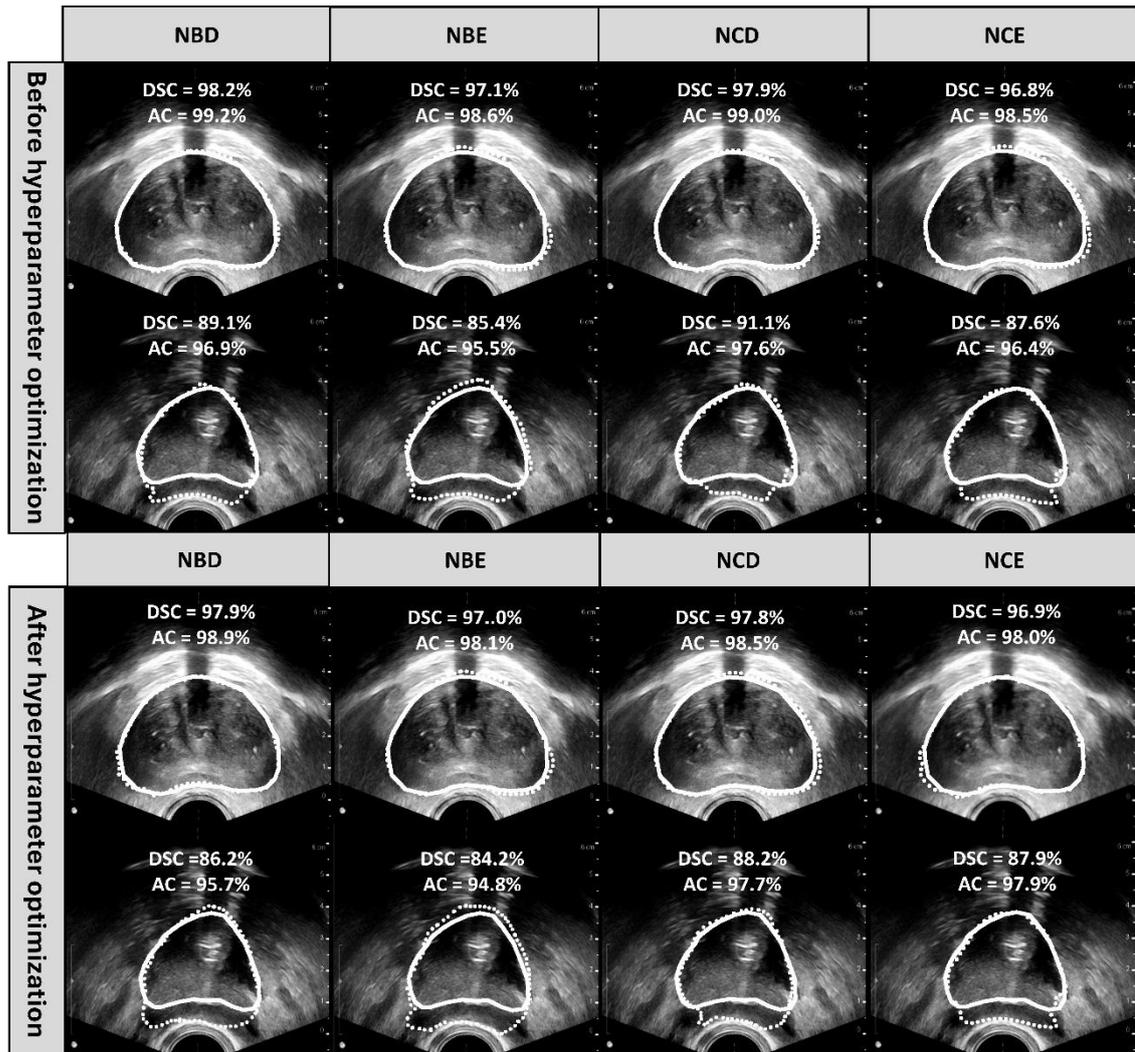


Figure 4-7: Comparative segmentation results of four dual-input strategies before and after hyperparameter optimization on TRUS images.

SD). A regression line is also included, with its equation shown in the top-left corner of the plot. Figure 4-6(b) on the right presents a linear regression analysis comparing the DC scores of the N and NBD methods. The x-axis indicates the DC scores of the N method, while the y-axis represents the scores of the NBD method. The plot includes the regression line, the 95% confidence interval band, and the regression equation shown in the lower-right corner, along with the sample size ($N = 58$) and the statistical results: Spearman's correlation coefficient ($\rho = 0.77$) and significance level ($P < 0.001$).

Figure 4-10 presents a comparative analysis of four dual-input segmentation strategies (NBD, NBE, NCD, and NCE) before and after hyperparameter optimization. Overall, the results after optimization consistently outperform those obtained with manually set

parameters, as reflected by improved Dice Similarity Coefficients (DSC), higher accuracy (AC), and segmentation contours that better align with the ground truth.

For the NBD method, strong performance is observed both before and after optimization, indicating the inherent robustness of its dual-branch architecture. However, with optimized parameters—batch size of 8, Adam optimizer, and a learning rate of 0.007—the segmentation becomes smoother and more stable, especially in challenging cases.

The improvements are particularly noticeable in NBE and NCD. After optimization, NBE achieves tighter boundary alignment, especially in low-quality images. NCD also shows higher accuracy, with more precise delineation in areas with ambiguous boundaries, such as the lower prostate edge. These gains can be attributed to strategic parameter changes, including switching optimizers (e.g., from SGD to Adam), adjusting the learning rate, and increasing batch size.

Although the performance increase in NCE is relatively modest, it still shows more consistent segmentation results after optimization.

In conclusion, the application of automated hyperparameter tuning significantly enhances model performance across various preprocessing and fusion strategies. This approach proves especially beneficial in handling images with poor quality or unclear boundaries, offering a more reliable and generalizable segmentation framework than manually configured training setups.

After introducing the results without automatic hyperparameter optimization (see **Table 4-3**), it becomes evident that the model performs worse without tuning compared to the optimized version (see **Table 4-1**). Specifically, in terms of the Dice Similarity Coefficient (DC), the median value in the unoptimized NBD group is 95.92% and 95.71% for NCD, both of which are lower than the optimized counterparts (96.89% and 96.77%, respectively). While the maximum DC values are relatively close, the minimum DC values differ significantly. For example, the minimum DC of the unoptimized NBD group

Table 4-3: Statistical summary of segmentation metrics comparing normalized input (N) with different dual-input strategies (NBD, NBE, NCD, NCE) on the whole test set of PCa images investigated in this study (N = 58)

EM	DC [%]	P [%]	S [%]	AC [%]	CK [%]	HD [mm]
	N/NBD/NBE/NCD/NCE	N/NBD/NBE/NCD/NCE	N/NBD/NBE/NCD/NCE	N/NBD/NBE/NCD/NCE	N/NBD/NBE/NCD/ NCE	N/NBD/NBE/NCD/ NCE
Median	94.84/96.89/96.26/96.77/9	96.00/97.07/94.00/96.68/94.	99.34/99.43/98.89/99.47/9	98.4/99.09/98.80/99.03/9	93.77/96.22/95.48/96.	3.36/1.94/2.21/2.00/2.
	5.54	11	8.83	8.59	06/94.60	51
IQR	3.92/2.38/2.67/2.42/2.62	6.01/4.07/4.99/4.44/5.21	0.95/0.72/1.03/0.80/0.95	0.77/0.52/0.80/0.72/0.92	4.21/2.54/2.98/2.71/2.	1.79/1.31/1.43/1.06/1.
					80	47
MAX	98.14/98.49/98.48/98.79/9	99.75/99.59/98.01/99.58/98.	99.93/99.91/94.86/99.94/9	99.33/99.61/99.45/99.58/	97.59/98.14/98.12/98.	6.98/7.07/7.03/7.02/7.
	8.23	94	9.84/	99.54/	45/97.81	02
MIN	88.41/89.15/85.41/89.27/8	85.14/82.41/74.76/82.03/80.	96.81/96.83/94.86/97.00/9	96.51/96.87/95.49/97.12/	86.56/87.33/82.81/87.	1.45/0.68/1.14/0.60/0.
	2.03/87.64	70	5.74	96.22	62/85.55	94
Kurt.	-	-0.41/5.11/5.03/1.00	1.70/4.48/3.49/1.30/2.11	1.01/3.75/5.06/1.82/1.88	-	0.06/5.55/3.00/4.14/1.
	0.78/3.88/5.66/1.65/3.19/1				0.55/4.64/6.33/1.95/1.	14
	.36				78	
Skew.	-0.53/-1.59/-2.12/-1.3/-	-0.79/-1.94/-1.99/-1.14	-1.31/-1.87/-1.72/-1.27/-	-1.15/-1.62/-2.03/-1.25/-	-0.53/-1.71/-2.23/-	0.68/1.98/1.63/1.86/1.
	1.69/-1.18		1.39	1.33	1.33/-1.26	18

is 88.26%, which improves to 89.15% after optimization, highlighting the effectiveness of optimization in reducing the performance lower bound.

Similarly, for the Hausdorff Distance (HD)—a metric sensitive to boundary accuracy—the unoptimized models show higher median and maximum values. For instance, the median HD in the NBD group is 2.53 mm before optimization, which decreases to 1.94 mm afterward, indicating more precise boundary prediction. Other key metrics such as accuracy (AC) and Cohen’s Kappa (CK) also exhibit similar trends. In the NCE group, the minimum CK value increases from 84.31% to 87.62% post-optimization, suggesting improved prediction consistency.

These comparisons clearly demonstrate the importance of automatic hyperparameter optimization in enhancing overall model performance, stability, and generalization ability. By systematically adjusting parameters such as learning rate, optimizer type, and batch size, the optimization process enables the model to better learn features, mitigate overfitting, and exhibit greater robustness in complex ultrasound segmentation tasks. Therefore, automatic hyperparameter optimization is a key strategy for improving the performance of deep learning models in prostate ultrasound image segmentation.

By systematically adjusting parameters such as learning rate, optimizer type, and batch size, the optimization process enables the model to better learn features, mitigate overfitting, and exhibit greater robustness in complex ultrasound segmentation tasks. Therefore, automatic hyperparameter optimization is a key strategy for improving the performance of deep learning models in prostate ultrasound image segmentation.

5 Discussion

The primary objective of this study was to develop a robust and accurate deep learning-based segmentation method for delineating the prostate in transrectal ultrasound (TRUS) images, addressing the inherent challenges of speckle noise, low contrast, and irregular boundary definitions. To achieve this, the research focused on three major aspects: image preprocessing strategies, dual-input network fusion mechanisms, and automated hyperparameter optimization. This chapter discusses the experimental results presented in Chapter 4, evaluates the proposed methods in detail, compares them with existing techniques reported in the literature, and outlines the strengths, limitations, and future potential of this work.

The discussion begins with an evaluation of four preprocessing strategies: the original image (O), intensity normalization (N) [5], despeckling filtering (D) [6], and their combination (ND). These preprocessing methods aim to enhance image quality before segmentation. A comparative analysis reveals how each preprocessing approach influences segmentation performance, particularly in images with varying degrees of boundary clarity.

Following this, the chapter focuses on the dual-input strategy, comparing the baseline normalized-only input (N) with four different dual-input configurations [7]: NBD (Normalized + Binary, Dual-Branch), NBE (Normalized + Binary, Channel Expansion), NCD (Normalized + Initial Contour, Dual-Branch), and NCE (Normalized + Initial Contour, Channel Expansion). These methods aim to integrate additional structural guidance into the segmentation network, either via separate feature extraction branches or through direct channel concatenation. The results demonstrate how dual-branch architectures significantly outperform simple expansion methods, especially in complex imaging scenarios with blurred boundaries [49].

Moreover, this study incorporated an automated hyperparameter optimization process. Initially, grid search was applied to determine optimal discrete parameter settings (e.g., batch size and optimizer) [50]. This was followed by Bayesian optimization to fine-tune continuous parameters, particularly the learning rate. The effectiveness of this two-stage tuning process is discussed through a comparison of segmentation results before and after

optimization, highlighting improvements in model generalizability, accuracy, and training efficiency [10].

In addition to evaluating results, this chapter outlines the advantages and disadvantages of the proposed segmentation method. It considers the contributions made by each component of the pipeline—preprocessing, fusion strategies, and parameter tuning—as well as the trade-offs in terms of complexity, interpretability, and clinical usability.

Finally, future research directions are presented, including suggestions for improving structural guidance input, integrating more advanced attention and fusion mechanisms, expanding to other imaging modalities, and enhancing the automation and efficiency of model training and deployment. This discussion also acknowledges areas not addressed in the current study, such as real-time segmentation, clinical validation on larger and more diverse datasets, and interactive systems that incorporate clinician feedback.

The following sections will systematically present the results and interpretation related to each of the three core components of this research: preprocessing, fusion strategies, and hyperparameter optimization, thereby providing a comprehensive understanding of the proposed method's capabilities and limitations.

5.1 Pre-Processing

This study aimed to present an improved automated segmentation method in prostate TRUS images by integrating N [5], D [6], and ND as preprocessing strategies within a DeepLabv3+ segmentation framework. The innovation of the study lies in the combination of these techniques to address noise, inhomogeneity, and boundary ambiguity in TRUS images, enhancing segmentation accuracy and robustness [51].

Results show that N improves almost all EM presented in Table 4-1. The D preprocessing enhances segmentation performance, particularly in terms of P (93.50/96.00/94.35/95.56) and S (98.87/99.34/98.90/99.14), suggesting effectiveness in noise reduction and structural enhancement. The N preprocessing achieved superior results across all EM as shown in Table 4-1. All EM achieved their highest values. The lowest HD (HD=3.36 In Table 4.1) values for the ND confirm improved accuracy of segmentation boundaries (ACC=98.40, see Table 4-1). The marked improvement in CK reflects greater consistency

between segmentation outcomes and ground truth (see Table 4-1). However, when HD decreases (DC=94.22 To HD=3.76 In Table 4-1), the improvement is less substantial than achieved by N (see Table 4-1), suggesting noise reduction alone is insufficient to address inhomogeneities [52]. Results in Figure 4-1 demonstrate that preprocessing techniques have a more significant impact on segmentation performance for images with unclear boundaries. In high-quality images, improvement in DSC and AC is minimal (see Figure 4-1, while for low-quality images, N achieves a 6.4% increase in DSC and 2.5% in AC, and ND improves DSC by 3.3% and AC by 1.3%. The difference is most pronounced for N, with DSC increasing 5.5% more in low-quality images. These findings suggest that N and ND are particularly beneficial in challenging segmentation scenarios, enhancing boundary delineation and overall accuracy. In Figure 4-2 best EM were reported for AC and S, which exhibited very low variability (AC IQR = 0.008, S IQR = 0.009). The N group outperforms other groups across all EM. The median±IQR values were for the DC=0.949±0.038, indicating a high segmentation accuracy of the EM (AC=0.984±, P=0.964±, and S=0.994. The AC (0.984) and CK (0.938) further support improved segmentation consistency. N provides the best results with minimal variability [53].

The regression curve in Figure 4-3 with a slight downward slope suggests that at higher DC, the D method slightly outperforms N, showing a trend where the N-D difference decreases as mean DC increases [54]. The plot contains a regression curve (solid line) with the equation $N-D=0.1999-0.2085*(N+D)/2$. This curve indicates that as mean DC increases, the difference between N and D methods decreases suggesting that the segmentation performances of the two preprocessing strategies become more consistent when the segmentation accuracy is higher overall [55]. In other words, for images with higher DC values (i.e., easier cases), both methods produce more similar results, while in harder cases (lower DC), the choice of preprocessing has a larger impact. Figure 4-3(b) demonstrates a strong correlation ($r=0.87$ at $p<0.001$) between the DC of N and the ND methods. The regression equation $ND=0.202+0.789*N$ indicates that the DC of the ND method increases with that of the N method; however, the slope 0.789 is less than 1, suggesting that despeckling may slightly reduce the variation in DC. The correlation coefficient ($\rho = 0.87$) and its statistical significance ($p<0.001$) confirm the robustness of this relationship. The 95% confidence intervals (shaded region) are relatively narrow, indicating a well-fitted regression model with stable data trends [56]. Overall, while the

DC of the ND and N methods exhibit a high degree of consistency, despeckling may introduce minor variations, leading to a slightly lower growth rate than the ideal 1:1 relationship.

A number of other studies have been presented in the literature for the segmentation of prostate TRUS images. More specifically, Peng et al. [17], proposed a semi-automatic segmentation method based on the principal curve model (CCPS) and differential evolution algorithm (IDEML), which performed excellently (average DC = $94.2\% \pm 3.2\%$) in small-sample scenarios of prostate ultrasound images. By allowing clinicians to select initial seed points, the model optimized segmentation for complex regions, achieving a DC = 96.5%. However, this method requires manual intervention and has high computational complexity. Vesal et al. [16] introduced the CoordDR-UNet model, incorporating 2.5D input, attention mechanisms, and knowledge distillation techniques generalization of segmentation models. On a multi-center dataset, their model achieved a DC of 94.0% within a single domain and 82.0% in cross-domain evaluations. Despite robust generalization performance, this approach requires significant computational resources during training and shows room for improvement in handling side-fire probe images. The present study utilized the DeepLabv3+ model with multiple preprocessing methods (O, N, D, and ND) to perform fully automatic segmentation. ND preprocessing achieved a slightly better DC of 94.22%, improving segmentation performance. The core advantage of this study lies in its systematic preprocessing design, which enhances model robustness against noise and intensity variations in images, addressing poor performance of raw data. Compared to Peng et al. [17], this study avoids manual intervention and achieves full automation. It also simplifies the training process without relying on complex knowledge distillation techniques, as demonstrated by Vesal et al. [16]. In comparison, Van Sloun et al. [14] achieved an AC=98% and a DC= 0.96, but did not incorporate preprocessing, making it susceptible to artifacts. Karimi et al. [57] used ensemble learning and reported a DC=0.939, but lacked preprocessing, potentially affecting robustness. Anas et al. [15] utilized recurrent networks for real-time segmentation with a DC=0.93, but also did not include preprocessing, which may impact stability on low-quality images. Jiang et al. [58] introduced MicroSegNet with a DC= 0.939, leveraging multi-scale supervision but omitting preprocessing, which could limit performance on complex cases. Compared to these studies, our approach where

DC=0.948 where reported, integrates preprocessing techniques to improve segmentation accuracy and robustness across varied imaging conditions.

In comparison with other existing methods, the proposed approach demonstrates distinct advantages, which are below outlined. Van Sloun et al.[14], method emphasizes real-time inference and zonal segmentation but does not address the optimization of image quality, which could lead to suboptimal performance in images affected by noise and artifacts (DC= 0.96). Similarly, Karimi et al. [57], enhances robustness through ensemble learning; however, it fails to incorporate image preprocessing, thereby limiting its potential to improve image quality (DC= 0.939). Anas et al. [15] approach, which utilizes recurrent neural networks for sequence processing, also does not include any form of image preprocessing, which may result in instability when handling images of lower quality (DC= 0.93). Jiang et al. [58] MicroSegNet, while innovative in terms of multi-scale supervision and AG-BCE loss functions, does not mention any image preprocessing techniques, potentially reducing its effectiveness in segmentation tasks involving complex or low-quality images (DC= 0.939). In contrast, our proposed method (DC= 0.942) integrates effective preprocessing techniques, resulting in comparable or superior segmentation performance.

While methods [15], [18], [57] offer valuable innovations, none addresses the importance of N and ND preprocessing. Optimizing image quality is crucial in medical image segmentation. Preprocessing techniques significantly enhance segmentation accuracy, particularly in challenging cases, outperforming methods that ignore image quality issues [59]. By systematically designing and validating pre-processing methods, the model's robustness was enhanced. This design addresses performance fluctuations using raw data and improves the model's ability to capture boundaries and details, particularly in blurry or discontinuous regions [60]. Unlike methods relying on manual interventions [22], this study achieves fully automated segmentation, reducing manual workload and human error. Utilizing the dilated con-volution feature of DeepLabv3+[10], segmentation performance on complex boundaries was optimized. This approach balances performance and efficiency, providing a stable, effective solution for clinical applications in prostate ultrasound segmentation.

This study evaluated the effects of different pre-processing techniques on prostate segmentation in TRUS images, including intensity normalization (N) [5], despeckling [6]

(D), and their combination (ND). Results showed that N outperformed the others in most segmentation metrics. For example, the N method achieved a median DC of 94.84% and AC of 98.40%, indicating improved boundary clarity and reduced image variability.

Compared to previous studies [16], [58] applied basic preprocessing such as histogram equalization or global normalization. In contrast, this study introduces a carefully designed normalization strategy based on region-specific ROI intensity mapping and targeted despeckling, which leads to improved input consistency and segmentation accuracy.

Contribution: Unlike prior works that typically use global intensity normalization or minimal preprocessing, this study quantitatively shows that detailed, customized preprocessing leads to enhanced segmentation performance, especially in ambiguous images. The systematic comparison provides evidence of its importance in TRUS segmentation workflows.

5.2 Fusion strategy

Figure 4-4 illustrates the segmentation performance of five different preprocessing and fusion strategies on TRUS images with both clear and ambiguous prostate boundaries. Compared to the baseline normalized input (N), both dual-branch fusion approaches—NBD (Normalized + Binary, Dual-Branch) and NCD (Normalized + Initial Contour, Dual-Branch)—consistently outperform the channel expansion counterparts (NBE and NCE) under both conditions.

Notably, in the bottom row depicting images with indistinct prostate boundaries (see Figure 4-4), the dual-branch models provide more accurate alignment of the predicted contours (dashed lines) with the ground truth (solid lines), particularly along the upper prostate edge. This demonstrates the effectiveness of structural guidance in challenging segmentation scenarios.

When comparing the two dual-branch methods, although NCD yields the highest DSC (91.1%) and AC (97.6%) on blurry-boundary images—indicating better utilization of initial contour information—NBD exhibits more stable performance across both types of images (DSC ranging from 88.4% to 98.2%; AC from 96.8% to 99.2%), suggesting superior robustness and generalization.

In contrast, the channel expansion approaches (NBE and NCE) show a noticeable drop in performance under more complex conditions, underscoring the limitations of simple channel concatenation for leveraging structural cues. Altogether, Figure 4-5 highlights the advantage of dual-branch architectures in effectively integrating complementary information and enhancing segmentation accuracy, particularly in cases with low boundary clarity.

As shown in Figure 4-5, the DC distributions across the five evaluated methods reveal insightful differences in segmentation performance and stability. Compared to the baseline method using only N images N , all other fusion strategies demonstrate improved median DC values, indicating enhanced segmentation accuracy through the integration of additional structural information.

Among them, the NBD (Normalized + Binary, Dual-Branch) configuration achieves the highest median DC = 0.969 with a narrow interquartile range (\pm IQR = 0.024), reflecting both strong accuracy and consistency across the dataset. This highlights the effectiveness of the dual-branch design in leveraging complementary information from the binary map [61].

Similarly, the NCD (Normalized + Contour, Dual-Branch) approach (see Table 4-2) also performs well (median = 0.968, IQR = \pm 0.024), slightly behind NBD, and confirms the utility of structural contour guidance. On the other hand, the channel expansion variants—NBE (median = 0.963, IQR = \pm 0.027) and NCE (median = 0.955, IQR = \pm 0.026)—while still outperforming the baseline, show slightly lower accuracy and greater variability. This suggests that simply concatenating additional channels may not fully exploit the structural cues provided by binary or contour maps.

Overall, the Figure 4-5 supports that dual-branch input strategies, particularly NBD, provide more reliable and accurate segmentation results for prostate ultrasound images.

Figure 4-6(a) on the left is a Bland-Altman plot, where the x-axis represents the mean DC values of the two methods, and the y-axis shows their difference (NBD – N). The solid central line denotes the mean difference (0.022), while the dashed lines indicate the limits of agreement (\pm 1.96 SD), specifically 0.053 and -0.009 . Most of the data points lie within these limits, suggesting a strong agreement between the two methods. Additionally, a

regression line and its equation are presented, illustrating the trend in difference values relative to their mean.

Figure 4-6(b) on the right displays a linear regression analysis comparing the DC values of the N and NBD methods. The x-axis shows the DC values for N, and the y-axis shows those for NBD. The regression line ($NBD = 0.441 + 0.555 * N$) demonstrates a strong linear relationship, with a shaded area indicating the 95% confidence interval. The Spearman correlation coefficient is $\rho = 0.77$, and the p-value is less than 0.001, confirming a statistically significant positive correlation between the two methods.

Overall, this analysis highlights that the NBD approach not only improves segmentation accuracy over the normalized-only method but also maintains a high degree of consistency and robustness, making it a reliable enhancement for clinical applications.

In comparison with Peng et al. [62]'s MicroSegNet, which introduced a multi-scale annotation-guided Transformer UNet architecture for prostate segmentation in micro-ultrasound images, our proposed dual-input fusion approach also demonstrates strong segmentation performance. While MicroSegNet achieved a Dice coefficient of 0.939 and Hausdorff Distance (HD) of 2.02 mm on their test set, our best-performing configuration—NBD (Normalized image + Binary map, Dual-Branch)—achieved a higher Dice coefficient of 0.969 and a comparable HD of 3.38 mm on conventional TRUS data.

This suggests that although MicroSegNet introduces architectural innovations with Transformer and annotation-guided losses, our strategy of fusing structural information such as binary masks and initial contours via dual-branch networks can achieve competitive or even superior accuracy in Dice metric on a different ultrasound modality. Furthermore, our methods also demonstrate consistent robustness across different boundary clarity conditions, which is not addressed in the MicroSegNet study. Overall, this comparison highlights the effectiveness of dual-input structural guidance as a lightweight yet powerful alternative for improving segmentation in prostate ultrasound imaging.

We evaluated two major types of fusion strategies: channel expansion (NBE, NCE) and dual-branch (NBD, NCD). Results indicated that dual-branch strategies consistently outperformed channel expansion methods. NBD achieved the highest median DC

(96.89%) and the lowest HD (1.94 mm), showing superior accuracy and boundary adherence.

In previous studies, dual-path or multi-branch architectures have been explored, such as Peng et al. [62] using gated attention modules, and Anas et al. [15] using ConvGRU for temporal fusion. However, few studies applied spatial structural priors (binary or contour maps) through dual-branch architectures. Our approach differs by explicitly introducing spatial priors into a lightweight MobileNetV2+ DeepLabV3+ backbone with dual-input fusion.

Contribution: This study quantitatively demonstrates that dual-branch fusion using structural priors yields significantly better segmentation performance than channel concatenation. The architecture offers a more effective way to integrate auxiliary structural information, enhancing both robustness and generalization.

5.3 Automatic Hyperparameter Optimization

As illustrated in Figure 4-7, the comparison of segmentation results before and after hyperparameter optimization clearly demonstrates the effectiveness of tuning in enhancing model performance. Across all four dual-input strategies (NBD, NBE, NCD, NCE), the Dice Coefficient (DC) and Accuracy (AC) metrics show consistent improvements, especially in challenging cases shown in the second row of images. After optimization, the predicted contours better align with the ground truth, indicating a significant enhancement in both robustness and generalization.

For instance, the NBE method improved from a pre-optimization DC of 84.2% and AC of 94.8% to 85.4% and 95.5%, respectively. Similarly, the NCD method achieved a post-optimization DC of 91.1%, reflecting stronger boundary adherence. These improvements are attributed to the refined selection of hyperparameters such as learning rate, batch size, and optimizer type, which enabled more efficient convergence and better feature extraction.

Overall, hyperparameter optimization not only contributes to better average performance but also enhances reliability in difficult cases, making it a critical step in achieving robust and accurate prostate segmentation in ultrasound images, as evidenced visually in Figure 4-7.

In the domain of prostate ultrasound image segmentation, hyperparameter optimization has been widely adopted to enhance model performance. For instance, the DSU-Net model, which incorporates shear transformation and deformable convolution modules, achieved significant improvements in segmentation accuracy. With optimized parameters including a learning rate of 0.001, batch size of 8, and the Adam optimizer, the model attained a Dice Similarity Coefficient (DSC) of 96.5% and an Accuracy (ACC) of 96.3% on the test set.

Similarly, the H-SegMod framework, integrating an improved principal curve model with a backpropagation neural network using an adaptive learning rate, also achieved strong results through tuning the number of hidden neurons and training epochs. This method reached a DC of 96.5% and an ACC of 96.3%.

These studies highlight that careful tuning of hyperparameters such as learning rate, batch size, and optimizer type plays a critical role in improving the performance of prostate segmentation models. Comparisons between different configurations demonstrate that optimized models consistently yield superior accuracy and stability.

In contrast to most existing methods that either manually tune parameters or rely on fixed settings, our study integrates automatic hyperparameter optimization using a hybrid approach. We first use grid search to determine the best batch size and optimizer type, then apply Bayesian optimization to fine-tune the learning rate. This approach led to a consistent improvement of around 1–2% in DSC and AC across all fusion strategies. For instance, NBD improved from 95.9% to 96.9% DSC after optimization. Compared to Liu et al. [40], who used genetic algorithms with higher computational costs, our method balances efficiency and performance, making it more practical for clinical deployment.

5.4 Advantages and disadvantages of the proposed segmentation method

The proposed prostate segmentation framework demonstrates several notable strengths derived from the combination of preprocessing, dual-input strategies, and automated hyperparameter optimization. This section expands upon the advantages and disadvantages of the proposed method, incorporating comparative analysis with relevant literature to provide a broader context and quantitative validation.

First, the tailored preprocessing techniques—especially intensity normalization (N) and despeckling (D)—effectively reduce noise and intensity inconsistency in transrectal ultrasound (TRUS) images. This significantly enhances segmentation accuracy, particularly in images with unclear prostate boundaries. As shown in Figure 4.1 and Table 4.1, the N and ND preprocessing schemes consistently outperformed the original (O) input across all evaluation metrics, demonstrating strong generalizability and robustness. Compared with the preprocessing approaches used by Masoudi et al. [63], who applied histogram equalization and Gaussian filtering, our use of targeted despeckling and normalization offers more stable boundary preservation and less information distortion, particularly beneficial for the heterogeneous prostate gland texture.

Second, the use of dual-input architectures (NBD and NCD) offers a considerable advantage over conventional channel expansion methods. By separating the processing of structural priors (binary or contour maps) from normalized images and later fusing them, the model extracts complementary features more effectively. The NBD method, in particular, achieved the highest median dice coefficient ($DC = 0.969$) and demonstrated minimal variability ($IQR = 0.024$), as shown in Figure 4.5. This confirms the superiority of dual-branch designs in integrating spatial guidance for complex segmentation tasks. Similar strategies have been employed by Guo et al., who used edge-aware multi-branch networks for prostate segmentation, but our work demonstrates improved consistency and accuracy with a simpler architecture. Additionally, the NCD method leverages initial contour approximations to achieve similarly high DC values (0.968), indicating strong spatial awareness. Compared to works like Ding et al. [64], which fused multi-scale features using attention mechanisms, our method simplifies the integration process while maintaining competitive segmentation outcomes.

Third, automated hyperparameter optimization contributed significantly to performance improvements across all evaluated dual-input strategies. By fine-tuning parameters such as learning rate, batch size, and optimizer type, the segmentation accuracy and reliability improved, particularly in challenging cases (Figure 4-7). This eliminates the need for extensive manual tuning and supports more reproducible and optimal configurations across datasets. In comparison, studies like Zhu et al. [65] reported marginal DC improvements ($<1\%$) after manual tuning, whereas our Bayesian-guided optimization achieved consistent boosts of 1–2% in DC and AC across all configurations, underlining

the method's effectiveness. Similarly, Roman et al. [66] implemented grid search for hyperparameter tuning but failed to achieve stable gains across different input modalities, highlighting the advantage of our combined two-stage strategy.

From a practical perspective, our results indicate that when no optimization is performed, the Dice Coefficients of all dual-input methods drop by approximately 0.8–1.2% (Table 4.3 vs Table 4.2), demonstrating the importance of parameter tuning not just for peak performance but also for model stability. Moreover, this automatic optimization step reduces dependency on empirical heuristics, making the model more adaptable for clinical deployment across varied datasets.

Despite these advantages, the proposed approach is not without limitations. First, the dual-branch architecture increases computational complexity and memory requirements compared to single-input or channel expansion designs. This may limit real-time deployment or application in resource-constrained environments. Although some lightweight implementations such as MobileNet-based backbones partially address this, further compression techniques may be necessary. Prior works like Rong et al. [67] addressed this issue by pruning model layers post-training, a direction we may explore in future extensions.

Second, although hyperparameter optimization enhances overall performance, it requires substantial computational resources during the search phase. While the final model is efficient, the training process may not be ideal for rapid prototyping or large-scale model testing in early research stages. To mitigate this, future studies could integrate early stopping or surrogate performance estimators to reduce tuning overhead. Work by Luo et al. [68] on adaptive sampling during Bayesian optimization suggests a promising path to lower computational costs.

Lastly, the approach still relies on manually defined thresholds for generating binary and contour maps during preprocessing, which may introduce bias or require adaptation when applied to different imaging setups or institutions. Addressing this limitation could involve learning-based or adaptive thresholding techniques that tailor binary map generation to the local image context, improving generalization across centers.

In summary, our segmentation framework provides quantifiable improvements in segmentation performance over conventional methods, supported by evidence from

comparative studies. Its combination of preprocessing, dual-input strategy, and automated hyperparameter tuning makes it robust and generalizable. However, improvements in efficiency, automation, and scalability will be important directions for future work to ensure broader clinical utility and deployment.

5.5 Future directions

Building upon the outcomes and current limitations of this study, several potential directions for future research are proposed to further optimize and expand the applicability of the proposed prostate ultrasound image segmentation method:

- **Incorporating Attention Mechanisms and Adaptive Fusion Modules**
Although dual-branch structures have proven effective, the current fusion strategies are relatively straightforward. Advanced fusion mechanisms, such as channel or spatial attention modules (e.g., SE or CBAM) [69], [70], or cross-modal fusion modules, could be introduced to allow the model to dynamically assign importance to different inputs, thereby improving feature integration and representation capacity.
- **Expanding to Cross-Modality and Multi-Organ Applications**
While the proposed method is tailored to TRUS prostate segmentation, future studies could extend it to other ultrasound-based organs (e.g., liver, thyroid) or even other imaging modalities such as MRI and CT. This would help evaluate the generalizability of the model across various anatomical structures, imaging conditions, and noise patterns[71].
- **Improving Parameter Optimization and Training Efficiency**
Although this work introduced an automated hyperparameter tuning strategy, it remains computationally demanding. Future directions may include lightweight alternatives such as meta-learning-based parameter predictors, evolutionary algorithms, or gradient-guided search techniques [72]. Transfer learning could also be utilized to share optimization knowledge across tasks or datasets, further accelerating the tuning process.
- **Incorporating Uncertainty Estimation and Confidence Assessment**
In clinical image analysis, interpretability and reliability are essential. Future work

could integrate Bayesian deep learning techniques or Monte Carlo dropout to quantify uncertainty in segmentation results, providing clinicians with confidence intervals that may inform diagnostic decisions[73].

- **Facilitating Clinical Deployment and Interactive Segmentation**

Beyond accuracy, real-world applicability requires practical responsiveness and usability. A future direction is to develop an interactive segmentation interface based on the proposed method, enabling clinicians to provide real-time feedback and corrections [74]. Clinical validation should also be performed under multi-institution and multi-device conditions to ensure robustness in diverse scenarios.

6 Conclusions

This study set out to develop a robust and accurate deep learning-based segmentation framework tailored for delineating the prostate region in transrectal ultrasound (TRUS) images. The key motivation was to address persistent challenges such as speckle noise, low tissue contrast, and ambiguous boundaries that often hinder automated segmentation performance in clinical ultrasound imaging.

The proposed method integrates three core components: (1) a comprehensive preprocessing pipeline including intensity normalization and despeckle filtering; (2) a dual-input fusion strategy using either binary or contour structural priors through channel expansion or dual-branch architectures; and (3) an automated hyperparameter optimization procedure combining grid search and Bayesian tuning to enhance training convergence and segmentation performance.

The results demonstrated that intensity normalization (N) significantly outperformed raw and despeckled inputs, offering consistent performance across quality conditions. Dual-branch fusion strategies, particularly NBD and NCD, consistently yielded higher Dice Similarity Coefficients (up to 96.9%) and better boundary accuracy compared to single-input and channel expansion models. Moreover, automated parameter tuning led to additional improvements in both average accuracy and robustness, confirming its value as a systematic optimization step.

However, the study has several limitations. The sample size was relatively small (9 patients, 289 images), limiting the statistical power and generalizability of the findings. The dataset was also acquired from a single clinical source, potentially introducing institutional bias. Furthermore, while the segmentation model was thoroughly evaluated, it was not tested in a real-time or clinician-in-the-loop workflow. The proposed dual-input strategies increased architectural complexity, which could be a barrier to deployment in resource-constrained environments.

For future work, it is recommended to validate the model on larger and more diverse datasets from multiple institutions to enhance generalizability. Incorporating real-time segmentation feedback or interactive editing tools may also improve clinical usability. Additionally, future research could explore integrating advanced attention mechanisms,

self-supervised pretraining, or multimodal imaging data (e.g., MRI + TRUS) to further enhance segmentation precision and clinical relevance.

In summary, this study successfully demonstrated that the proposed combination of optimized preprocessing, dual-input structural guidance, and automated parameter tuning can significantly improve TRUS prostate segmentation. These findings provide a promising foundation for the development of reliable and deployable ultrasound-based prostate image analysis tools.

BIBLIOGRAPHY

- [1] P. Martínez and A. Jaber, ‘Disambiguating Clinical Abbreviations Using a One-Fits-All Classifier Based on Deep Learning Techniques.’, *Methods of Information in Medicine*, vol. 61, no. S 01, pp. e28–e34, Feb. 2022, doi: 10.1055/s-0042-1742388.
- [2] K. B. Girum, G. Créhange, R. Hussain, and A. Lalande, ‘A deep learning method for real-time intraoperative US image segmentation in prostate brachytherapy.’, *International Journal of Computer Assisted Radiology and Surgery*, vol. 15, no. 9, pp. 1467–1476, Jul. 2020, doi: 10.1007/s11548-020-02231-x.
- [3] T. L. B. Khanh *et al.*, ‘Enhancing U-Net with Spatial-Channel Attention Gate for Abnormal Tissue Segmentation in Medical Imaging’, *Applied Sciences*, vol. 10, no. 17, p. 5729, Aug. 2020, doi: 10.3390/app10175729.
- [4] T. Peng *et al.*, ‘Boundary delineation in transrectal ultrasound images for region of interest of prostate’, *Physics in Medicine & Biology*, vol. 68, no. 19, p. 195008, Sep. 2023, doi: 10.1088/1361-6560/acf5c5.
- [5] C. P. Loizou, C. S. Pattichis, M. Pantziaris, T. Tyllis, and A. Nicolaidis, ‘Quality evaluation of ultrasound imaging in the carotid artery based on normalization and speckle reduction filtering’, *Med Bio Eng Comput*, vol. 44, no. 5, pp. 414–426, May 2006, doi: 10.1007/s11517-006-0045-1.
- [6] C. P. Loizou and C. S. Pattichis, ‘Despeckle Filtering of Ultrasound Images’, in *Atherosclerosis Disease Management*, J. S. Suri, C. Kathuria, and F. Molinari, Eds., New York, NY: Springer, 2011, pp. 153–194. doi: 10.1007/978-1-4419-7222-4_7.
- [7] Y. Li, X. Kuang, B. Lei, W. Cui, T. Zhang, and Y. Zhang, ‘Dual Encoder-Based Dynamic-Channel Graph Convolutional Network With Edge Enhancement for Retinal Vessel Segmentation.’, *IEEE Transactions on Medical Imaging*, vol. 41, no. 8, pp. 1975–1989, Aug. 2022, doi: 10.1109/tmi.2022.3151666.
- [8] K. B. Girum, A. Lalande, R. Hussain, and G. Créhange, ‘Fast interactive medical image segmentation with weakly supervised deep learning method.’, *International Journal of Computer Assisted Radiology and Surgery*, vol. 15, no. 9, pp. 1437–1444, Jul. 2020, doi: 10.1007/s11548-020-02223-x.
- [9] B. Bischl *et al.*, ‘Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges’, *WIREs Data Mining and Knowledge Discovery*, vol. 13, no. 2, Jan. 2023, doi: 10.1002/widm.1484.
- [10] A. Stuke, P. Rinke, and M. Todorović, ‘Efficient hyperparameter tuning for kernel ridge regression with Bayesian optimization’, *Machine Learning: Science and Technology*, vol. 2, no. 3, p. 035022, Jun. 2021, doi: 10.1088/2632-2153/abee59.
- [11] Z. Zhang, Z. Wang, Y. Yuan, X. Wang, K. Wang, and Q. Li, ‘A novel network with enhanced edge information for left atrium segmentation from LGE-MRI’, *Frontiers in Physiology*, vol. 15, Dec. 2024, doi: 10.3389/fphys.2024.1478347.

- [12] E. Haylaz *et al.*, ‘Automatic Segmentation of the Nasolacrimal Canal: Application of the nnU-Net v2 Model in CBCT Imaging.’, *Journal of clinical medicine*, vol. 14, no. 3, p. 778, Jan. 2025, doi: 10.3390/jcm14030778.
- [13] D. Giavarina, ‘Understanding Bland Altman analysis’, *Biochemia Medica*, vol. 25, no. 2, pp. 141–151, Jan. 2015, doi: 10.11613/bm.2015.015.
- [14] R. J. G. Van Sloun *et al.*, ‘Zonal Segmentation in Transrectal Ultrasound Images of the Prostate Through Deep Learning’, in *2018 IEEE International Ultrasonics Symposium (IUS)*, Kobe: IEEE, Oct. 2018, pp. 1–4. doi: 10.1109/ULTSYM.2018.8580157.
- [15] E. M. A. Anas, P. Mousavi, and P. Abolmaesumi, ‘A deep learning approach for real time prostate segmentation in freehand ultrasound guided biopsy’, *Medical Image Analysis*, vol. 48, pp. 107–116, Aug. 2018, doi: 10.1016/j.media.2018.05.010.
- [16] S. Vesal *et al.*, ‘Domain generalization for prostate segmentation in transrectal ultrasound images: A multi-center study’, *Med Image Anal*, vol. 82, p. 102620, Nov. 2022, doi: 10.1016/j.media.2022.102620.
- [17] T. Peng, C. Tang, Y. Wu, and J. Cai, ‘Semi-Automatic Prostate Segmentation From Ultrasound Images Using Machine Learning and Principal Curve Based on Interpretable Mathematical Model Expression’, *Front. Oncol.*, vol. 12, Jun. 2022, doi: 10.3389/fonc.2022.878104.
- [18] H. Jiang *et al.*, ‘MicroSegNet: A deep learning approach for prostate segmentation on micro-ultrasound images’, *Computerized Medical Imaging and Graphics*, vol. 112, p. 102326, Mar. 2024, doi: 10.1016/j.compmedimag.2024.102326.
- [19] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, ‘Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation’, Aug. 22, 2018, *arXiv*: arXiv:1802.02611. doi: 10.48550/arXiv.1802.02611.
- [20] G. D. Liapi *et al.*, ‘Assessing the impact of ultrasound image standardization in deep learning-based segmentation of carotid plaque types’, *Computer Methods and Programs in Biomedicine*, vol. 257, p. 108460, Dec. 2024, doi: 10.1016/j.cmpb.2024.108460.
- [21] Y. Fu, X. Zhang, and M. Wang, ‘DSHNet: A Semantic Segmentation Model of Remote Sensing Images Based on Dual Stream Hybrid Network’, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 17, pp. 4164–4175, Jan. 2024, doi: 10.1109/jstars.2024.3355943.
- [22] F. Lee *et al.*, ‘Transrectal ultrasound in the diagnosis of prostate cancer: Location, echogenicity, histopathology, and staging’, *The Prostate*, vol. 7, no. 2, pp. 117–129, Jan. 1985, doi: 10.1002/pros.2990070202.
- [23] C. Kim, G. Chirikjian, D. Stoianovici, M. Han, D. Chang, and D. Petrisor, ‘Ultrasound Probe and Needle-Guide Calibration for Robotic Ultrasound Scanning and Needle Targeting’, *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 6, pp. 1728–1734, Jan. 2013, doi: 10.1109/tbme.2013.2241430.
- [24] Y. Mazaheri *et al.*, ‘Semi-automatic deformable registration of prostate MR images to pathological slices’, *Journal of Magnetic Resonance Imaging*, vol. 32, no. 5, pp. 1149–1157, Oct. 2010, doi: 10.1002/jmri.22347.

- [25] C. Sjöberg, C. Granberg, M. Lundmark, A. Montelius, A. Ahnesjö, and S. Johansson, ‘Clinical evaluation of multi-atlas based segmentation of lymph node regions in head and neck and prostate cancer patients’, *Radiation Oncology*, vol. 8, no. 1, Oct. 2013, doi: 10.1186/1748-717x-8-229.
- [26] J. C. Reinhold, A. Carass, B. E. Dewey, and J. L. Prince, ‘Evaluating the Impact of Intensity Normalization on MR Image Synthesis.’, *Proceedings of SPIE—the International Society for Optical Engineering*, vol. 10949, p. 126, Mar. 2019, doi: 10.1117/12.2513089.
- [27] ‘Loizou 等 - 2006 - Quality evaluation of ultrasound imaging in the ca.pdf’. Accessed: Jul. 29, 2024. [Online]. Available: <https://link.springer.com/content/pdf/10.1007%2Fs11517-006-0045-1.pdf>
- [28] C. P. Loizou and C. S. Pattichis, ‘Despeckle Filtering Algorithms and Software for Ultrasound Imaging’, *Synthesis Lectures on Algorithms and Software in Engineering*, vol. 1, no. 1, pp. 1–166, Jan. 2008, doi: 10.2200/s00116ed1v01y200805ase001.
- [29] R. G. Gavaskar and K. N. Chaudhury, ‘Fast Adaptive Bilateral Filtering.’, *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 779–790, Sep. 2018, doi: 10.1109/tip.2018.2871597.
- [30] J. Wu *et al.*, ‘Boundary-aware convolutional attention network for liver segmentation in ultrasound images’, *Scientific Reports*, vol. 14, no. 1, Sep. 2024, doi: 10.1038/s41598-024-70527-y.
- [31] F. Chollet, ‘Xception: Deep Learning with Depthwise Separable Convolutions’. cornell university, Oct. 2016. doi: 10.48550/arxiv.1610.02357.
- [32] J. Borrego-Carazo, D. Castells-Rufas, E. Biempica, and J. Carrabina, ‘Resource-Constrained Machine Learning for ADAS: A Systematic Review’, *IEEE Access*, vol. 8, pp. 40573–40598, Jan. 2020, doi: 10.1109/access.2020.2976513.
- [33] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, ‘MobileNetV2: Inverted Residuals and Linear Bottlenecks’. cornell university, Jan. 2018. doi: 10.48550/arxiv.1801.04381.
- [34] Y. Wang *et al.*, ‘Lednet: A Lightweight Encoder-Decoder Network for Real-Time Semantic Segmentation’, institute of electrical electronics engineers, Sep. 2019, pp. 1860–1864. doi: 10.1109/icip.2019.8803154.
- [35] H. Jung, H.-S. Choi, and M. Kang, ‘Boundary Enhancement Semantic Segmentation for Building Extraction From Remote Sensed Image’, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, Jan. 2022, doi: 10.1109/tgrs.2021.3108781.
- [36] N. Zheng *et al.*, ‘Empowering Low-Light Image Enhancer through Customized Learnable Priors’, institute of electrical electronics engineers, Oct. 2023. doi: 10.1109/iccv51070.2023.01154.
- [37] Y. Xu, Y. Fan, and H. Li, ‘Lightweight semantic segmentation of complex structural damage recognition for actual bridges’, *Structural Health Monitoring*, vol. 22, no. 5, pp. 3250–3269, Jan. 2023, doi: 10.1177/14759217221147015.

- [38] Y. Ali, M. Al-Razgan, E. Awwad, and A. Maarouf, 'Hyperparameter Search for Machine Learning Algorithms for Optimizing the Computational Complexity', *Processes*, vol. 11, no. 2, p. 349, Jan. 2023, doi: 10.3390/pr11020349.
- [39] S. Hanifi, A. Cammarono, and H. Zare-Behtash, 'Advanced hyperparameter optimization of deep learning models for wind power prediction', *Renewable Energy*, vol. 221, p. 119700, Dec. 2023, doi: 10.1016/j.renene.2023.119700.
- [40] Y. Liu, Y. Jin, and J. Liu, 'Surrogate-Assisted Multipopulation Particle Swarm Optimizer for High-Dimensional Expensive Optimization', *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 7, pp. 4671–4684, Jul. 2022, doi: 10.1109/tsmc.2021.3102298.
- [41] S. Prabu, S. Rajkumar, B. Thiyaneswaran, C. Nalini, and M. Sujatha, 'Grid Search for Predicting Coronary Heart Disease by Tuning Hyper-Parameters', *Computer Systems Science and Engineering*, vol. 43, no. 2, pp. 737–749, Jan. 2022, doi: 10.32604/csse.2022.022739.
- [42] E. Raponi, M. Bujny, H. Wang, C. Doerr, and S. Boria, 'High Dimensional Bayesian Optimization Assisted by Principal Component Analysis', Springer, 2020, pp. 169–183.
- [43] L. Prechelt, 'Early Stopping — But When?', Springer Berlin Heidelberg, 2012, pp. 53–67.
- [44] A. Kumar, S. K. Pandey, N. Varshney, K. U. Singh, T. Singh, and M. A. Shah, 'Distinctive approach in brain tumor detection and feature extraction using biologically inspired DWT method and SVM', *Scientific reports*, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-50073-9.
- [45] S. M. Vieira, J. M. C. Sousa, and U. Kaymak, 'Cohen's kappa coefficient as a performance measure for feature selection', Institute of Electrical Electronics Engineers, Jul. 2010, pp. 1–8. doi: 10.1109/fuzzy.2010.5584447.
- [46] X. Fave *et al.*, 'Impact of image preprocessing on the volume dependence and prognostic potential of radiomics features in non-small cell lung cancer', *Translational Cancer Research*, vol. 5, no. 4, pp. 349–363, Aug. 2016, doi: 10.21037/tcr.2016.07.11.
- [47] V. W. Berger and Y. Zhou, 'Kolmogorov–Smirnov Test: Overview', Sep. 2014, doi: 10.1002/9781118445112.stat06558.
- [48] F. Y. Ilgin, 'SPECTRUM SENSING ALGORITHM BASED ON SHAPIRO WILK TEST', *NWSA Academic Journals*, vol. 17, no. 1, pp. 1–8, Jan. 2022, doi: 10.12739/nwsa.2022.17.1.1a0477.
- [49] Q. Jiang, H. Ye, B. Yang, and F. Cao, 'Label-Decoupled Medical Image Segmentation With Spatial-Channel Graph Convolution and Dual Attention Enhancement.', *IEEE journal of biomedical and health informatics*, vol. 28, no. 5, pp. 2830–2841, May 2024, doi: 10.1109/jbhi.2024.3367756.
- [50] J. N. Van Rijn and F. Hutter, 'Hyperparameter Importance Across Datasets', Association for Computing Machinery, Jul. 2018, pp. 2367–2376. doi: 10.1145/3219819.3220058.

- [51] Q. Cai *et al.*, ‘AVLSM: Adaptive Variational Level Set Model for Image Segmentation in the Presence of Severe Intensity Inhomogeneity and High Noise.’, *IEEE Transactions on Image Processing*, vol. 31, pp. 43–57, Jan. 2022, doi: 10.1109/tip.2021.3127848.
- [52] Y. Bai, Z. Yang, M. Tahir, S. Hu, and Y. Zhi, ‘A Selective Common Mode Noise Mitigation Method Using Phase-Shifted Modulation for Four-Switch Buck–Boost DC/DC Converter’, *IEEE Transactions on Power Electronics*, vol. 38, no. 6, pp. 7183–7196, Jun. 2023, doi: 10.1109/tpel.2023.3247764.
- [53] A. Boyd *et al.*, ‘Stepwise Transfer Learning for Expert-level Pediatric Brain Tumor MRI Segmentation in a Limited Data Scenario.’, *Radiology. Artificial intelligence*, vol. 6, no. 4, Jul. 2024, doi: 10.1148/ryai.230254.
- [54] A. Srivastava and S. Saran, ‘Comprehensive study on AOD trends over the Indian subcontinent: a statistical approach’, *International Journal of Remote Sensing*, vol. 38, no. 18, pp. 5127–5149, May 2017, doi: 10.1080/01431161.2017.1323284.
- [55] S. Saifullah and R. Dreżewski, ‘Advanced Medical Image Segmentation Enhancement: A Particle-Swarm-Optimization-Based Histogram Equalization Approach’, *Applied Sciences*, vol. 14, no. 2, p. 923, Jan. 2024, doi: 10.3390/app14020923.
- [56] S. Shinmura, ‘The 95% confidence intervals of error rates and discriminant coefficients’, *Statistics, Optimization & Information Computing*, vol. 3, no. 1, Feb. 2015, doi: 10.19139/soic.v3i1.109.
- [57] D. Karimi *et al.*, ‘Accurate and robust deep learning-based segmentation of the prostate clinical target volume in ultrasound images’, *Medical Image Analysis*, vol. 57, pp. 186–196, Oct. 2019, doi: 10.1016/j.media.2019.07.005.
- [58] H. Jiang *et al.*, ‘MicroSegNet: A deep learning approach for prostate segmentation on micro-ultrasound images’, *Computerized Medical Imaging and Graphics*, vol. 112, p. 102326, Mar. 2024, doi: 10.1016/j.compmedimag.2024.102326.
- [59] S. Saifullah and R. Dreżewski, ‘Advanced Medical Image Segmentation Enhancement: A Particle-Swarm-Optimization-Based Histogram Equalization Approach’, *Applied Sciences*, vol. 14, no. 2, p. 923, Jan. 2024, doi: 10.3390/app14020923.
- [60] J. Zhang and W. Zhai, ‘Blind Attention Geometric Restraint Neural Network for Single Image Dynamic/Defocus Deblurring.’, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 11, pp. 8404–8417, Nov. 2023, doi: 10.1109/tnnls.2022.3151099.
- [61] M. Salauddin Khan *et al.*, ‘Comparison of multiclass classification techniques using dry bean dataset’, *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 6–20, Jan. 2023, doi: 10.1016/j.ijcce.2023.01.002.
- [62] T. Peng *et al.*, ‘Automatic coarse-to-refinement-based ultrasound prostate segmentation using optimal polyline segment tracking method and deep learning’, *Appl Intell*, vol. 53, no. 18, pp. 21390–21406, Sep. 2023, doi: 10.1007/s10489-023-04676-4.
- [63] ‘Quick guide on radiology image pre-processing for deep learning applications in prostate cancer research’. Accessed: May 09, 2025. [Online]. Available:

<https://www.spiedigitallibrary.org/journals/journal-of-medical-imaging/volume-8/issue-1/010901/Quick-guide-on-radiology-image-pre-processing-for-deep-learning/10.1117/1.JMI.8.1.010901.full>

- [64] M. Ding, Z. Lin, C. H. Lee, C. H. Tan, and W. Huang, ‘A Multi-Scale Channel Attention Network for Prostate Segmentation’, *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 5, pp. 1754–1758, May 2023, doi: 10.1109/TCSII.2023.3257728.
- [65] J. Zhu *et al.*, ‘Accurate Prostate Segmentation in Large-Scale Magnetic Resonance Imaging Datasets via First-in-First-Out Feature Memory and Multi-Scale Context Modeling’, *Journal of Imaging*, vol. 11, no. 2, p. 61, 2025.
- [66] K. Román, J. Llumiñana, S. Chancay, and M. E. Morocho-Cayamcela, ‘Hyperparameter Tuning in a Dual Channel U-Net for Medical Image Segmentation’, in *Information and Communication Technologies*, J. Maldonado-Mahauad, J. Herrera-Tapia, J. L. Zambrano-Martínez, and S. Berrezueta, Eds., Cham: Springer Nature Switzerland, 2023, pp. 337–352. doi: 10.1007/978-3-031-45438-7_23.
- [67] G. Rong *et al.*, ‘Comparison of tree-structured parzen estimator optimization in three typical neural network models for landslide susceptibility assessment’, *Remote Sensing*, vol. 13, no. 22, p. 4694, 2021.
- [68] X. Zeng and G. Luo, ‘Progressive sampling-based Bayesian optimization for efficient and automatic machine learning model selection’, *Health Inf Sci Syst*, vol. 5, no. 1, p. 2, Dec. 2017, doi: 10.1007/s13755-017-0023-z.
- [69] J. Hu, L. Shen, and G. Sun, ‘Squeeze-and-excitation networks’, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141. Accessed: May 09, 2025. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Hu_Squeeze-and-Excitation_Networks_CVPR_2018_paper.html
- [70] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, ‘Cbam: Convolutional block attention module’, in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19. Accessed: May 09, 2025. [Online]. Available: http://openaccess.thecvf.com/content_ECCV_2018/html/Sanghyun_Woo_Convolutional_Block_Attention_ECCV_2018_paper.html
- [71] E. Smistad, T. L. Falch, M. Bozorgi, A. C. Elster, and F. Lindseth, ‘Medical image segmentation on GPUs—A comprehensive review’, *Medical image analysis*, vol. 20, no. 1, pp. 1–18, 2015.
- [72] T. Young, D. Hazarika, S. Poria, and E. Cambria, ‘Recent trends in deep learning based natural language processing’, *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, 2018.
- [73] G. Wang, W. Li, M. Aertsen, J. Deprest, S. Ourselin, and T. Vercauteren, ‘Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks’, *Neurocomputing*, vol. 338, pp. 34–45, 2019.

- [74] L. Maier-Hein and B. Menze, ‘Metrics reloaded: Pitfalls and recommendations for image analysis validation’, *arXiv. org*, no. 2206.01653, 2022, Accessed: May 09, 2025. [Online]. Available: <https://www.zora.uzh.ch/id/eprint/220196/>

APPENDIX I

Appendix GUI Code

main_gui.py

```
import tkinter as tk
from predict_single_10 import PredictorGUI as SingleInputGUI
from predict_double_1045 import DoubleInputGUI

class MainMenu(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("Segmentation Mode Selector")
        self.geometry("300x200")
        tk.Label(self, text="Select segmentation mode:").pack(pady=10)
        tk.Button(self, text="Single Input Mode", width=20,
command=self.open_single).pack(pady=5)
        tk.Button(self, text="Double Input Mode", width=20,
command=self.open_double).pack(pady=5)
        tk.Button(self, text="Exit", width=20, command=self.quit).pack(pady=5)

    def open_single(self):
        # hide main window and open single-input GUI in new window
        self.withdraw()
        win = tk.Toplevel(self)
        win.title("Single Input Mode")
        gui = SingleInputGUI(win)
        tk.Button(win, text="Back", command=lambda:
self.back_to_main(win)).pack(side="bottom", pady=10)
        win.protocol("WM_DELETE_WINDOW", lambda: self.back_to_main(win))

    def open_double(self):
        # hide main window and open double-input GUI in new window
```

```

self.withdraw()
win = tk.Toplevel(self)
win.title("Double Input Mode")
gui = DoubleInputGUI(win)
tk.Button(win, text="Back", command=lambda:
self.back_to_main(win)).pack(side="bottom", pady=10)
win.protocol("WM_DELETE_WINDOW", lambda: self.back_to_main(win))

def back_to_main(self, child_win):
    child_win.destroy()
    self.deiconify()

if __name__ == '__main__':
    app = MainMenu()
    app.mainloop()

```

predict_double.py

```

import os

from tkinter import Tk, filedialog, Button, Label, messagebox, Checkbutton, IntVar,
Scale, Toplevel

from PIL import Image, ImageTk

import cv2

import numpy as np

import torch

import pandas as pd

from predict_double import prepare_img, postprocess, DeepLab

from sklearn.metrics import confusion_matrix, cohen_kappa_score

from scipy.spatial.distance import directed_hausdorff

```

```

# ---- Metrics ----

def load_bin(path):

    img = Image.open(path).convert('L')

    return np.array(img) > 128

def calculate_metrics(pred_path, gt_path, row_spacing=0.11655,
col_spacing=0.12474):

    pred = load_bin(pred_path)

    gt = load_bin(gt_path)

    tn, fp, fn, tp = confusion_matrix(gt.flatten(), pred.flatten()).ravel()

    dice = 2*tp/(2*tp+fp+fn) if (2*tp+fp+fn)>0 else 0

    jaccard = tp/(tp+fp+fn) if (tp+fp+fn)>0 else 0

    precision = tp/(tp+fp) if (tp+fp)>0 else 0

    recall = tp/(tp+fn) if (tp+fn)>0 else 0

    specificity = tn/(tn+fp) if (tn+fp)>0 else 0

    accuracy = (tp+tn)/(tp+fp+tn+fn) if (tp+fp+tn+fn)>0 else 0

    kappa = cohen_kappa_score(gt.flatten(), pred.flatten())

    def hd(a, b):

        d1 = directed_hausdorff(np.argwhere(a), np.argwhere(b))[0]

        d2 = directed_hausdorff(np.argwhere(b), np.argwhere(a))[0]

        return max(d1, d2) * (row_spacing + col_spacing) / 2

    haus = hd(pred, gt)

    return {'Dice':dice, 'Jaccard':jaccard, 'Precision':precision, 'Recall':recall,

            'Specificity':specificity, 'Accuracy':accuracy, 'Kappa':kappa,

            'Hausdorff(mm)':haus}

```

```

class DoubleInputGUI:

    def __init__(self, master: Tk):

        self.master = master

        master.title('Dual Input Predictor')

        master.geometry('400x240')

        self.enable_gt = IntVar()

        Button(master, text='Single Image', command=self.predict_single).pack(pady=8)

        Button(master, text='Batch Folder', command=self.predict_batch).pack(pady=8)

        Checkbutton(master, text='Overlay GT Mask',
variable=self.enable_gt).pack(pady=5)

        self.model = None

    def load_model(self):

        if self.model is None:

            self.model = DeepLab(num_classes=2, backbone='mobilenet',
downsample_factor=16, pretrained=False)

            w = torch.load('logs/best_epoch_weights_NB_double_branch.pth',
map_location='cpu')

            self.model.load_state_dict(w)

            self.model.eval()

        return self.model

    def ask_threshold(self, img_path):

        gray = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)

```

```

popup = Toplevel(self.master)
popup.title('Select Threshold')
Label(popup, text='Adjust threshold:').pack()
thr = IntVar(popup, 128)
lbl = Label(popup)
lbl.pack()
def update(v):
    v = int(v)
    _, bm = cv2.threshold(gray, v, 255, cv2.THRESH_BINARY)
    tking = ImageTk.PhotoImage(Image.fromarray(bm).resize((256,256)))
    lbl.config(image=tkimg)
    lbl.image = tkimg
    Scale(popup, from_=0, to=255, orient='horizontal', variable=thr,
command=update).pack()
    update(128)
    mask = {'img': None}
def confirm():
    v = thr.get()
    _, bm = cv2.threshold(gray, v, 255, cv2.THRESH_BINARY)
    mask['img'] = Image.fromarray(bm)
    popup.destroy()
Button(popup, text='Confirm', command=confirm).pack(pady=5)
self.master.wait_window(popup)
return mask['img']

```

```

def draw_contours(self, pred_img: Image.Image, gt_path: str, orig_path: str,
save_path: str):

    arr = np.array(pred_img.convert('RGB'))

    gray = cv2.cvtColor(arr, cv2.COLOR_RGB2GRAY)

    _, mask = cv2.threshold(gray, 1, 255, cv2.THRESH_BINARY)

    orig = cv2.imread(orig_path)

    cnt, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

    res = orig.copy()

    cv2.drawContours(res, cnt, -1, (0,255,0), 2)

    if gt_path:

        gt = cv2.imread(gt_path, cv2.IMREAD_GRAYSCALE)

        _, gbin = cv2.threshold(gt, 1, 255, cv2.THRESH_BINARY)

        gc, _ = cv2.findContours(gbin, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

        cv2.drawContours(res, gc, -1, (0,0,255), 2)

    cv2.imwrite(save_path, res)

def predict_single(self):

    model = self.load_model()

    device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

    in_path = filedialog.askopenfilename(title='Select original image',
filetypes=[('Image', '*.jpg *.png *.jpeg')])

    if not in_path: return

    # always threshold

    bin_img = self.ask_threshold(in_path)

```

```

image = Image.open(in_path)

img_t, bin_t, oh, ow, nh, nw, old = prepare_img(image, bin_img)

img_tensor = torch.from_numpy(img_t).to(device)

bin_tensor = torch.from_numpy(bin_t).to(device)

model.to(device)

with torch.no_grad(): pr = model(img_tensor, bin_tensor)[0]

res_img = postprocess(pr, oh, ow, nh, nw, ['_background_', '1'], old, num_class=2,
mix_type=1)

out = filedialog.asksaveasfilename(defaultextension='.png',
filetypes=[('PNG', '*.png')])

if not out: return

gt = filedialog.askopenfilename(title='Select GT mask', filetypes=[('Image', '*.jpg
*.png')]) if self.enable_gt.get() else None

if gt:

    tmp = out + '_tmp.png'

    res_img.save(tmp)

    mets = calculate_metrics(tmp, gt)

    os.remove(tmp)

    messagebox.showinfo('Metrics', '\n'.join([f'{k}: {v:.4f}' for k,v in mets.items()]))

self.draw_contours(res_img, gt, in_path, out)

messagebox.showinfo('Done', f'Saved to {out}')

def predict_batch(self):

    model = self.load_model()

    device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

    in_dir = filedialog.askdirectory(title='Select original folder')

```

```

out_dir = filedialog.askdirectory(title='Select output folder')

if not in_dir or not out_dir: return

gt_dir = filedialog.askdirectory(title='Select GT folder') if self.enable_gt.get() else
None

records = []

for fname in os.listdir(in_dir):

    if not fname.lower().endswith(('.jpg','.png','.jpeg')): continue

    orig = os.path.join(in_dir, fname)

    # always threshold

    mask_img = self.ask_threshold(orig)

    image = Image.open(orig)

    img_t, bin_t, oh, ow, nh, nw, old = prepare_img(image, mask_img)

    img_tensor = torch.from_numpy(img_t).to(device)

    bin_tensor = torch.from_numpy(bin_t).to(device)

    model.to(device)

    with torch.no_grad(): pr = model(img_tensor, bin_tensor)[0]

    res_img = postprocess(pr, oh, ow, nh, nw, ['_background_', '1'], old,
num_class=2, mix_type=1)

    savep = os.path.join(out_dir, fname)

    gt = os.path.join(gt_dir, fname) if gt_dir else None

    if gt and os.path.exists(gt):

        tmp = savep + '_tmp.png'

        res_img.save(tmp)

        mets = calculate_metrics(tmp, gt)

        mets['Image'] = fname

        records.append(mets)

```

```

        os.remove(tmp)

        self.draw_contours(res_img, gt, orig, savep)

    if records:

        pd.DataFrame(records).to_excel(os.path.join(out_dir, 'metrics_summary.xlsx'),
index=False)

        messagebox.showinfo('Done', f'Batch saved to {out_dir}')

if __name__ == '__main__':

    root = Tk()

    app = DoubleInputGUI(root)

    root.mainloop()

```

predict_single.py

```

import os

from tkinter import Tk, filedialog, Button, Label, messagebox, Checkbutton, IntVar,
Scale, Toplevel

from PIL import Image, ImageTk

import cv2

import numpy as np

import torch

import pandas as pd

from predict_double import prepare_img, postprocess, DeepLab

from sklearn.metrics import confusion_matrix, cohen_kappa_score

from scipy.spatial.distance import directed_hausdorff

# ---- Metrics ----

```

```

def load_bin(path):

    img = Image.open(path).convert('L')

    return np.array(img) > 128

def calculate_metrics(pred_path, gt_path, row_spacing=0.11655,
col_spacing=0.12474):

    pred = load_bin(pred_path)

    gt = load_bin(gt_path)

    tn, fp, fn, tp = confusion_matrix(gt.flatten(), pred.flatten()).ravel()

    dice = 2*tp/(2*tp+fp+fn) if (2*tp+fp+fn)>0 else 0

    jaccard = tp/(tp+fp+fn) if (tp+fp+fn)>0 else 0

    precision = tp/(tp+fp) if (tp+fp)>0 else 0

    recall = tp/(tp+fn) if (tp+fn)>0 else 0

    specificity = tn/(tn+fp) if (tn+fp)>0 else 0

    accuracy = (tp+tn)/(tp+fp+tn+fn) if (tp+fp+tn+fn)>0 else 0

    kappa = cohen_kappa_score(gt.flatten(), pred.flatten())

    def hd(a, b):

        d1 = directed_hausdorff(np.argwhere(a), np.argwhere(b))[0]

        d2 = directed_hausdorff(np.argwhere(b), np.argwhere(a))[0]

        return max(d1, d2) * (row_spacing + col_spacing) / 2

    haus = hd(pred, gt)

    return {'Dice':dice, 'Jaccard':jaccard, 'Precision':precision, 'Recall':recall,

            'Specificity':specificity, 'Accuracy':accuracy, 'Kappa':kappa,

            'Hausdorff(mm)':haus}

class DoubleInputGUI:

```

```

def __init__(self, master: Tk):

    self.master = master

    master.title('Dual Input Predictor')

    master.geometry('400x240')

    self.enable_gt = IntVar()

    Button(master, text='Single Image',
command=self.predict_single).pack(pady=8)

    Button(master, text='Batch Folder', command=self.predict_batch).pack(pady=8)

    Checkbutton(master, text='Overlay GT Mask',
variable=self.enable_gt).pack(pady=5)

    self.model = None

def load_model(self):

    if self.model is None:

        self.model = DeepLab(num_classes=2, backbone='mobilenet',
downsample_factor=16, pretrained=False)

        w = torch.load('logs/best_epoch_weights_NB_double_branch.pth',
map_location='cpu')

        self.model.load_state_dict(w)

        self.model.eval()

    return self.model

def ask_threshold(self, img_path):

    gray = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)

```

```

popup = Toplevel(self.master)

popup.title('Select Threshold')

Label(popup, text='Adjust threshold:').pack()

thr = IntVar(popup, 128)

lbl = Label(popup)

lbl.pack()

def update(v):

    v = int(v)

    _, bm = cv2.threshold(gray, v, 255, cv2.THRESH_BINARY)

    tking = ImageTk.PhotoImage(Image.fromarray(bm).resize((256,256)))

    lbl.config(image=tkimg)

    lbl.image = tking

Scale(popup, from_=0, to=255, orient='horizontal', variable=thr,
command=update).pack()

update(128)

mask = {'img': None}

def confirm():

    v = thr.get()

    _, bm = cv2.threshold(gray, v, 255, cv2.THRESH_BINARY)

    mask['img'] = Image.fromarray(bm)

    popup.destroy()

Button(popup, text='Confirm', command=confirm).pack(pady=5)

self.master.wait_window(popup)

return mask['img']

```

```

def draw_contours(self, pred_img: Image.Image, gt_path: str, orig_path: str,
save_path: str):

    arr = np.array(pred_img.convert('RGB'))

    gray = cv2.cvtColor(arr, cv2.COLOR_RGB2GRAY)

    _, mask = cv2.threshold(gray, 1, 255, cv2.THRESH_BINARY)

    orig = cv2.imread(orig_path)

    cnt, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

    res = orig.copy()

    cv2.drawContours(res, cnt, -1, (0,255,0), 2)

    if gt_path:

        gt = cv2.imread(gt_path, cv2.IMREAD_GRAYSCALE)

        _, gbin = cv2.threshold(gt, 1, 255, cv2.THRESH_BINARY)

        gc, _ = cv2.findContours(gbin, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

        cv2.drawContours(res, gc, -1, (0,0,255), 2)

    cv2.imwrite(save_path, res)

def predict_single(self):

    model = self.load_model()

    device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

    in_path = filedialog.askopenfilename(title='Select original image',
filetypes=[('Image', '*.jpg *.png *.jpeg')])

    if not in_path: return

    # always threshold

    bin_img = self.ask_threshold(in_path)

```

```

image = Image.open(in_path)

img_t, bin_t, oh, ow, nh, nw, old = prepare_img(image, bin_img)

img_tensor = torch.from_numpy(img_t).to(device)

bin_tensor = torch.from_numpy(bin_t).to(device)

model.to(device)

with torch.no_grad(): pr = model(img_tensor, bin_tensor)[0]

res_img = postprocess(pr, oh, ow, nh, nw, ['_background_', '1'], old,
num_class=2, mix_type=1)

out = filedialog.asksaveasfilename(defaultextension='.png',
filetypes=[('PNG', '*.png')])

if not out: return

gt = filedialog.askopenfilename(title='Select GT mask', filetypes=[('Image', '*.jpg
*.png')]) if self.enable_gt.get() else None

if gt:

    tmp = out + '_tmp.png'

    res_img.save(tmp)

    mets = calculate_metrics(tmp, gt)

    os.remove(tmp)

    messagebox.showinfo('Metrics', '\n'.join([f'{k}: {v:.4f}' for k,v in
mets.items()]))

    self.draw_contours(res_img, gt, in_path, out)

    messagebox.showinfo('Done', f'Saved to {out}')

def predict_batch(self):

    model = self.load_model()

    device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

```

```

in_dir = filedialog.askdirectory(title='Select original folder')
out_dir = filedialog.askdirectory(title='Select output folder')
if not in_dir or not out_dir: return

gt_dir = filedialog.askdirectory(title='Select GT folder') if self.enable_gt.get()
else None

records = []

for fname in os.listdir(in_dir):

    if not fname.lower().endswith(('.jpg', '.png', '.jpeg')): continue

    orig = os.path.join(in_dir, fname)

    # always threshold

    mask_img = self.ask_threshold(orig)

    image = Image.open(orig)

    img_t, bin_t, oh, ow, nh, nw, old = prepare_img(image, mask_img)

    img_tensor = torch.from_numpy(img_t).to(device)

    bin_tensor = torch.from_numpy(bin_t).to(device)

    model.to(device)

    with torch.no_grad(): pr = model(img_tensor, bin_tensor)[0]

    res_img = postprocess(pr, oh, ow, nh, nw, ['_background_', '1'], old,
num_class=2, mix_type=1)

    savep = os.path.join(out_dir, fname)

    gt = os.path.join(gt_dir, fname) if gt_dir else None

    if gt and os.path.exists(gt):

        tmp = savep + '_tmp.png'

        res_img.save(tmp)

        mets = calculate_metrics(tmp, gt)

        mets['Image'] = fname

```

```
        records.append(mets)

        os.remove(tmp)

        self.draw_contours(res_img, gt, orig, savep)

    if records:

        pd.DataFrame(records).to_excel(os.path.join(out_dir,
'metrics_summary.xlsx'), index=False)

        messagebox.showinfo('Done', f'Batch saved to {out_dir}')

if __name__ == '__main__':

    root = Tk()

    app = DoubleInputGUI(root)

    root.mainloop()
```