Proceedings of the 3rd International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/2025.LD30738

Improving Surgical Tool Segmentation under Bleeding Corruption via Specialized Augmentation Strategy

Yicheng Shao

Culver Academies, Culver, USA
662635@culver.org

Abstract. Artificial intelligence (AI) shows great potential for improving surgical efficiency, precision, and autonomy in surgical robotic systems. However, the robustness of deep learning-based algorithms remains a critical challenge as the surgical environments shows much variance in real application. Most deep learning-based segmentation models, though highly effective on benchmarking datasets, often fail during unforeseen non-adversarial corruptions such as occlusions, bleeding, or low brightness. In this study, we introduce a domain-specific augmentation strategy to enhance model robustness against possible surgical corruptions that is not seen in the training data. Our method simulates key corruptions, including blood simulation, brightness adjustment, and contrast adjustment. Based on the SegSTRONG-C benchmark, we evaluate a baseline U-Net model on a binary surgical tool segmentation task. While the baseline shows strong performance on clean images, its accuracy drops substantially on the corrupted test data. Incorporating our proposed augmentations significantly improves performance on corrupted inputs while preserving accuracy on the clean domain. These findings underscore the importance of specific augmentation for model's robustness and demonstrate a practical pathway toward more reliable and generalizable segmentation models for real-world surgical robotics applications.

Keywords: intelligent surgical robotics, surgical AI, deep learning, model robustness, surgical tool segmentation.

1. Introduction

Surgical robot is taking more important role in healthcare, due to their unique advantages, including high precision, better range of motion, minimal invasion of the human body, and possibility of remote operation. Meanwhile artificial intelligence (AI) algorithms also presents a rapid advancement. Thus, integrating AI into surgical robotic systems has become a key area of research. This integration aims to enhance the intelligence of robotic systems, improving surgical efficiency and success rates. Although many image segmentation algorithms have achieved remarkable success in general computer vision tasks [1–6], they often rely on large scale training data and high-quality image inputs. However, such ideal conditions are difficult to guarantee in surgical environments due to the complexity and variability of real-world operations [7, 8]. Surgical scenes may involve less reliable lighting, occlusions, tissue deformation, and limited camera perspectives. These conditions always result in low image quality and consistency. Furthermore, unexpected complications like smoke, bleeding, can

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2025.LD30738

happen during actual procedures but these events are rarely captured during data collection or simulation, causing significant challenges to the robustness and generalization of segmentation algorithms. As a result, designing algorithms capable of accurately identifying and localizing surgical tools under those corrupted scenes is both essential and challenging.

The key focus of this study is to explore algorithms that enables widely applied deep learning architectures to correctly segment surgery tools in corrupted images during inference, while only clean data is provided during training.

While prior work has attempted to improve robustness through standard data augmentations or synthetic datasets, such approaches did not make specific design for complications of real surgical scenes that can be expected. In contrast, our strategy directly targets these issues by simulating specific corruptions as customized data augmentation like blood simulation, brightness adjustment, and contrast adjustment. This enables the model to learn more realistic visual representations of surgical tools under these corruptions, offering a effective and practical solution to enhance segmentation performance in real-world surgical applications.

To test the effectiveness of the specialized data augmentation, this work adapts the settings from SegStrongC benchmark that holds a separate dataset to check the model's performance in corrupted images. For this project, we focus especially on images partly covered by blood, which is a common complication the model will have to face during surgery. We apply the dataset from the SegSTRONG-C[7] challenge to conduct a study on the binary surgical tool segmentation task. We report the DICE[9] score as the quantitative metric for evaluation. We apply UNet[2] as a representative of the baseline networks and progressively add customized data augmentation, including color jittering and our blood occlusion simulation, into the training pipeline. The results show that while the baseline UNet achieves strong performance on clean images, it suffers a significant drop when tested on corrupted data (DICE $\approx 51\%$). Incorporating color jittering helps recover part of this loss, but the most notable improvement comes from our blood-occlusion simulation, which raises the corrupted-test performance above $> 70\%$ while maintaining high scores on the clean set. These findings demonstrate that domain-specific augmentations are critical for improving robustness and reliability in surgical tool segmentation.

2. Related Work

Image segmentation is foundational task in computer vision. In the recent years, deep learning–based approaches achieving remarkable success. In the context of surgical robotics, accurate segmentation of surgical tools is essential for real-time navigation, automation, and safety assurance. Methods including architectures such as U-Net[2], DeepLab[10], and Mask R-CNN[11] have been proposed to address this task and shown strong performance under clean, controlled benchmarking imaging conditions. However, their robustness under real-world surgical conditions remains questionable due to their limited generalization to out-of-distribution scenarios.

Recent studies have validated the gap between the performance of segmentation models under ideal conditions and visually corrupted environments. In particular, the SegSTRONG-C[7] challenge introduces a curated dataset of surgical tool images subjected to realistic, non-adversarial corruptions such as smoke, bleeding, and low brightness. This work demonstrates that even state-of-the-art models, if not trained on these corruptions, suffer significant performance drops, revealing a crucial limitation in existing approaches. CaRTS addresses the robustness via a new causal model including the robot kinematics as auxiliary data [12, 13]

While prior work has attempted to improve robustness through standard augmentations or synthetic datasets [14–17], such approaches did not take the prior knowledge of the expected compli-

DOI: 10.54254/2755-2721/2025.LD30738

cations of real surgical scenes. In contrast, our work proposes a targeted augmentation strategy that introduces surgery-specific corruptions into clean training data. By simulating scenes such as bleeding variation, we aim to proactively prepare models for the expected types of visual degradation they will face in practice. This complements existing benchmarks by addressing robustness at the training stage and offering a practical, domain-specific solution without requiring extensive real-world corrupted data.

3. Method

To address the issue of model overfitting to clean, ideal training data, we introduce a series of handcrafted data augmentations that simulate the visual challenges present in real surgical environments. These augmentations include synthetic effects such as simulated blood stains, color diffusion, and brightness alterations. The goal is to expose the model to more realistic conditions during training so it can better recognize surgical tools affected by environmental factors such as blood occlusion, low illumination, or partial visual obstruction.

3.1. Network Architecture

To perform binary segmentation of surgical tools from endoscopic video frames, we adopt a U-Net-based architecture tailored for high-resolution spatial localization and robustness to visual variation. The network follows a classic encoder-decoder structure with symmetric skip connections, enabling the combination of coarse semantic information from deeper layers with fine-grained localization cues from shallower layers.

3.1.1. Encoder (Contracting Path)

The encoder is composed of five convolutional stages that progressively reduce the spatial dimensions while increasing the channel depth. Each stage consists of one layer with either ReLU[18] or LeakyReLU[18] activations, optionally followed by Batch Normalization. The encoder blocks are constructed using the helper function `add_conv_stage`, which allows toggling batch normalization with the `useBN` flag.

- **Conv1:** Input image of size $[B, 3, H, W]$ is mapped to $[B, 32, H, W]$.
- **Conv2 to Conv5:** Each stage is followed by a 2×2 max pooling operation, halving the spatial resolution at each level, resulting in progressively smaller feature maps with deeper representations: $[64, H/2, W/2]$, $[128, H/4, W/4]$, $[256, H/8, W/8]$, and $[512, H/16, W/16]$.

3.1.2. Decoder (Expanding Path)

The decoder mirrors the encoder with four upsampling stages, each consisting of a transposed convolution followed by a single convolutional block. The upsampled feature maps are concatenated with the corresponding encoder outputs via skip connections to recover spatial detail lost during downsampling.

- **Upsample54:** Upsamples $[512, H/16, W/16]$ to $[256, H/8, W/8]$.
- **Conv4m:** Merges upsampled features with encoder output from `Conv4` and outputs $[256, H/8, W/8]$.
- **Upsample43** $\rightarrow$ **Conv3m:** Repeats the process to $[128, H/4, W/4]$.
- **Upsample32** $\rightarrow$ **Conv2m:** Produces $[64, H/2, W/2]$.
- **Upsample21** $\rightarrow$ **Conv1m:** Restores spatial resolution to $[32, H, W]$.

DOI: 10.54254/2755-2721/2025.LD30738

3.1.3. Final Prediction Layer

The final prediction is generated by a 1×1 convolution that reduces the channel dimension to 1, followed by a sigmoid activation to produce a pixel-wise probability map:

$$\text{Output} = \sigma(\text{Conv}_{1\times1}(x))$$

where σ denotes the sigmoid function. The output is a binary mask of shape $[B, 1, H, W]$ indicating the predicted presence of surgical tools at each pixel.

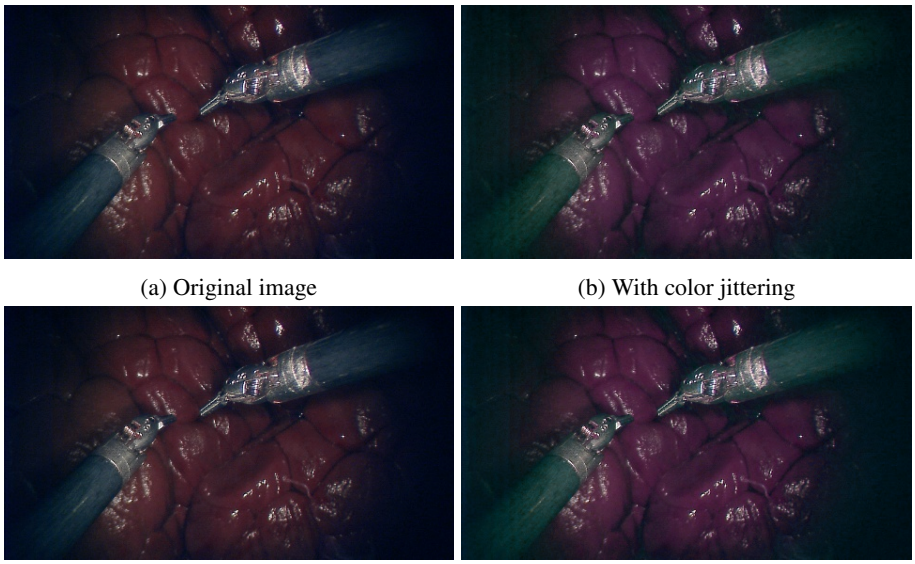
3.1.4. Implementation Details

All convolutional layers use kernel size 3, stride 1, and padding 1 unless otherwise specified. Transposed convolutions use kernel size 4, stride 2, and padding 1 to ensure proper spatial alignment during upsampling. The network supports optional Batch Normalization for improved convergence stability. This architecture allows the model to capture both low-level high resolution features and high-level semantic features.

3.2. Augmentation Algorithm

In our training pipeline, we design and apply the following data augmentation techniques to enhance the model's robustness against unseen but expected complications in the surgical scene.

– **Color Jittering:** We employ the ColorJitter module from the torchvision library to introduce random variations in brightness, contrast, saturation, and hue. This operation changes the color distribution of the input by adjusting pixel intensities in the HSV space, effectively simulating changes in lighting variability. By repeatedly train the model to these color–augmented image of the same scene, the network learns to rely less on raw pixel appearance and more on structural and textural features of the surgical tools. This increases robustness to differences in lighting and camera settings.



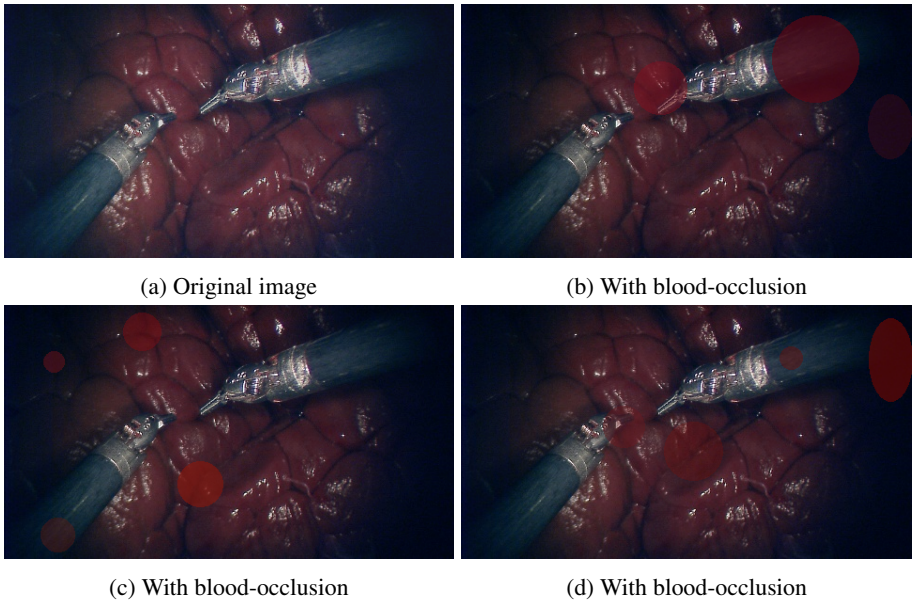
(a) Original image (b) With color jittering

(c) With color jittering (d) With color jittering

Figure 1: Example of the color jittering augmentation used during training.

DOI: 10.54254/2755-2721/2025.LD30738

– **Blood Simulation:** To simulate occlusion effects from surgical bleeding, we use the Python Imaging Library (PIL)[19] to overlay semi–transparent, randomly placed ellipses of varying radii and shades of red onto training images. Each ellipse is added with a alpha channel with the underlying pixels using an alpha range of 0.2–0.6, producing realistic partial occlusions of tool regions. Importantly, ellipse centers are sampled to focus on tool areas, making the augmentation targeted rather than background–only. This compels the model to infer tool boundaries and presence even when parts are obscured by blood or debris, thereby improving segmentation robustness under conditions of partial visibility.



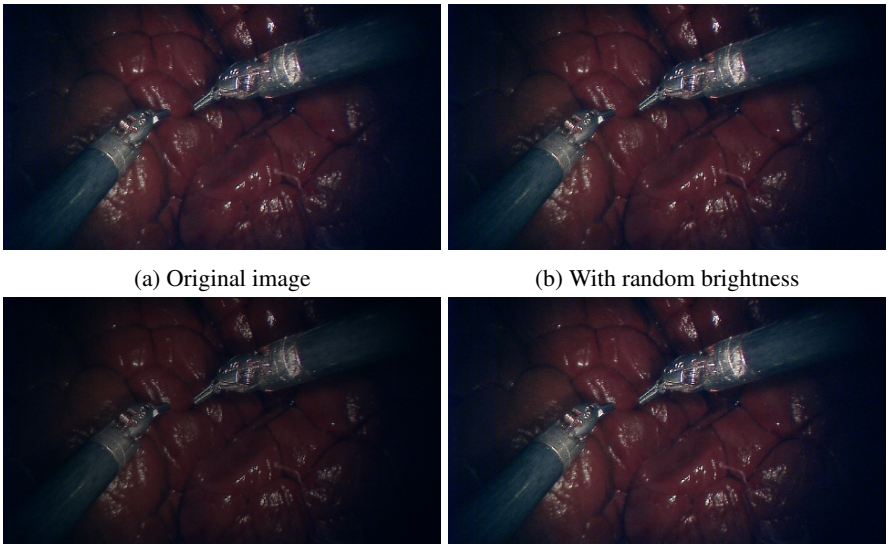
(a) Original image (b) With blood-occlusion

(c) With blood-occlusion (d) With blood-occlusion

Figure 2: Example of the blood-occlusion augmentation used during training.

– **Random Brightness Adjustment:** Using PIL's brightness enhancer, we apply multiplicative scaling of all pixel intensities by a factor sampled from a uniform range. This augmentation mimics real–world variability in lighting, such as sudden dimming when the light source is blocked, or oversaturation when the camera is too close to reflective surfaces. By training on images that vary from underexposed to overexposed, the network develops invariance to global illumination shifts and learns to focus on edge–based cues and structural consistency instead of absolute brightness, which is highly unstable in surgical video.

DOI: 10.54254/2755-2721/2025.LD30738

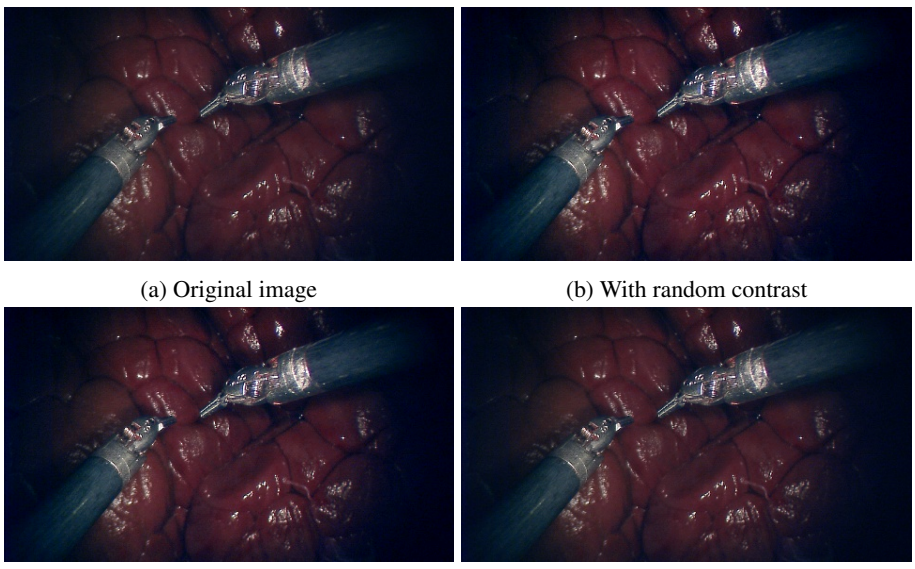


(a) Original image (b) With random brightness

(c) With random brightness (d) With random brightness

Figure 3: Example of the brightness augmentation used during training.

– **Random Contrast Adjustment:** We adjust the dynamic range of input images using PIL's contrast enhancer[19], which scales pixel values relative to the mean intensity. Depending on the sampled factor, this operation can flatten the image (low contrast) or exaggerate differences (high contrast). Such changes replicate conditions encountered during endoscopy, such as reduced visibility due to fogging or increased sharpness from specular highlights. By learning from both extremes, the model becomes more capable of distinguishing tool boundaries even in visually degraded frames, increasing resilience to camera artifacts and scene–dependent contrast variation.



(a) Original image (b) With random contrast

(c) With random contrast (d) With random contrast

Figure 4: Example of the contrast augmentation used during training.

DOI: 10.54254/2755-2721/2025.LD30738

By training the model on this artificially modified data, we improve its ability to generalize to real-world surgical scenarios, enabling it to detect and localize tools even under adverse visual conditions, rather than relying solely on recognition of ”clean” or idealized training samples.

4. Experiment

4.1. Dataset

To evaluate the robustness of surgical tool segmentation models under realistic visual perturbations, we use the SegSTRONG-C dataset [7], a curated benchmark specifically designed to assess performance under non-adversarial corruptions. The dataset is released as a sub-challenge under EndoVis 2024.

4.1.1. Base Data

The base dataset contains mock endoscopic video sequences recorded using two patient-side manipulators (PSMs) from the da Vinci surgical robot, operating in a controlled environment with animal tissue backgrounds to ensure photo-realism. For each video, trajectories were generated via manual teleoperation, and binary segmentation masks of the surgical tools were created using a semi-automated annotation pipeline. The data was collected using stereo endoscopic cameras and recorded at 10 frames per second.

Each sequence contains 300 frames per camera (left and right), and in total, the dataset includes 17 sequences captured under different robot and camera configurations. Annotation masks were generated through a multi-step process: background subtraction was used to generate prompts for the Segment Anything Model (SAM)[20], followed by expert manual correction of failure cases to ensure label accuracy.

4.1.2. Corruption Design

To simulate real-world intraoperative conditions, SegSTRONG-C[7] applies four sets of **non-adversarial** corruptions to the original images. by replaying the same robotic trajectories under modified environmental conditions:

- **Background Change:** Different types of animal tissue were used to alter background appearance.
- **Smoke:** Artificial fog was introduced using a fog machine to simulate surgical smoke.
- **Bleeding:** Fake blood was applied to obscure tool visibility, simulating occlusion due to bleeding.
- **Low Brightness:** The endoscopic light source is reduced.

Corruptions were applied to each test sequence by replaying the same robotic kinematics, ensuring the corrupted and clean sequences are aligned. This setup enables rigorous assessment of model robustness under a range of real-world visual corruptions. Unlike synthetic methods, generate more realistic appearance.

4.1.3. Challenge Design

The SegSTRONG-C benchmark[7] focuses on **robust generalization**, where models are trained solely on clean data and evaluated on unseen corrupted versions. This setup mimics real-world conditions where annotated, corrupted data may be scarce or unavailable. The benchmark reports performance using metrics such as the Dice coefficient, under both clean and corrupted domains.

DOI: 10.54254/2755-2721/2025.LD30738

4.2. Evaluation Metric

To quantitatively evaluate the segmentation performance of our model, we adopt a combined loss function during training and use the Dice coefficient as the primary evaluation metric during testing. This choice is motivated by the fact that the Dice score is particularly well-suited for imbalanced segmentation tasks, such as surgical tool detection, where the foreground (i.e., the tool) typically occupies a much smaller region than the background.

4.2.1. Dice Coefficient

The Dice coefficient, also known as the F1-score for segmentation, measures the overlap between the predicted binary mask and the ground truth mask. It is defined as:

$$\text{Dice}(P, G) = \frac{2|P \cap G|}{|P| + |G|}$$

where P denotes the predicted mask, G denotes the ground truth mask. In our implementation, predictions are thresholded at 0.5 to obtain binary outputs, and thxe Dice score is computed per sample and averaged across the test set.

4.2.2. Loss Function

During training, we employ a hybrid loss function that combines Binary Cross-Entropy (BCE) loss with Dice loss:

$$\mathcal{L}_{\text{total}} = \frac{1}{2}\mathcal{L}_{\text{BCE}} + \frac{1}{2}\mathcal{L}_{\text{Dice}}$$

– **Binary Cross-Entropy (BCE) Loss:** Measures pixel-wise classification error between predicted probabilities and ground truth labels, treating each pixel independently. It is defined as:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$$

where N is the number of pixels, $y_i \in \{0, 1\}$ is the ground truth label for pixel i, and $p_i \in (0, 1)$ is the predicted probability for that pixel.

– **Dice Loss:** Directly optimizes the region-level overlap between the predicted and ground truth masks. It is defined as:

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2\sum_i p_i g_i + \epsilon}{\sum_i p_i + \sum_i g_i + \epsilon}$$

where p_i and g_i represent the predicted and ground truth values at each pixel i, respectively.

This combined loss function optimize both pixel-level accuracy and region-level overlap.

4.3. Implementation Details

The U-Net model is implemented in PyTorch. Training is conducted with a batch size of 10 and a fixed learning rate of 1×10^{-3}. We apply the Adam optimizer with default momentum parameters ($\beta_1 = 0.9, \beta_2 = 0.999$). The network is trained for 10 epochs. All input images and corresponding masks are resized to 256×256 before training, ensuring that the spatial dimensions are divisible by 16

DOI: 10.54254/2755-2721/2025.LD30738

to fit the downsampling and upsampling requirements. After resizing, images are converted to tensors in the predefined normalized range $[0, 1]$ to fit the numerical scope of the network, which stabilizes gradient updates during optimization. The final network layer applies a 1×1 convolution followed by a `sigmoid` activation. To obtain binary predictions aligned with the ground-truth masks, a hard threshold of 0.5 is applied.

To ensure reproducibility, all random operations within the augmentation pipeline and network initialization are seeded. The same preprocessing and normalization pipeline was applied consistently during training and testing, with the only difference being the inclusion of domain-specific augmentations during training.

4.4. Main Results

Vanilla baseline. The baseline U–Net model, trained without augmentations, achieved strong convergence with steadily decreasing training loss from 0.1758 (epoch 0) to 0.0451 (epoch 9). Validation loss decreased until epoch 4 (0.0815) before plateauing around 0.083 at later epochs. Under clean test conditions, the vanilla model achieved a high DICE score (~ 0.945). However, when evaluated on corrupted images (e.g., with blood occlusion or color jittering), performance dropped sharply to ~ 0.512, demonstrating limited robustness to visual disturbances.

Augmentation with brightness, contrast, color jitter, and occlusion (Train1). When training with the full augmentation suite (brightness, contrast, color jitter, and occlusion), the model initially experienced slower convergence and higher validation loss compared to the baseline (validation: 0.214 at epoch 1, later stabilizing near 0.108). Nonetheless, robustness improved: clean test DICE score remained strong (0.929), and corrupted performance rose markedly to ~ 0.735, showing that targeted occlusions and photometric changes significantly reduce brittleness to realistic disturbances.

Augmentation with color jitter and occlusion (Train2). The combination of color jitter and occlusion yielded the best balance. Training loss converged smoothly from 0.1477 to 0.0480, while validation loss decreased from 0.1010 (epoch 0) to 0.0823 (epoch 9). On evaluation, the model preserved a high clean DICE score (0.931) and achieved a corrupted DICE score of ~ 0.714, slightly below the full augmentation setup but still vastly better than the vanilla baseline. This indicates that occlusion-based augmentation is the dominant contributor to robustness, while additional photometric variation provides marginal benefits.

Augmentation with color jitter only (Train3). Augmented with color jittering only, the model achieved a high clean DICE score (0.912), but the corrupted DICE score was noticeably lower (0.652), reflecting less resilience to severe occlusions. Training and validation curves also showed higher variance, where validation loss rises to 0.213 before stabilizing near 0.117. This suggests that photometric changes alone cannot generate similar effect that the occlusion simulation has.

Summary of findings. Overall, augmentations substantially improved robustness to surgical disturbances at only minor cost to clean performance. The vanilla baseline excelled on clean data but failed under corruptions. By contrast, models trained with occlusion-based augmentations (Train1 and Train2) generalized far better to corrupted test conditions, maintaining 70–73% performance compared to only 51% for the baseline. These results highlight the necessity of domain-specific data augmentations—particularly occlusion modeling—for reliable surgical tool segmentation under intraoperative variability.

DOI: 10.54254/2755-2721/2025.LD30738

Table 1: Cross–comparison of U–Net models trained with different augmentation strategies. Reported are the final validation loss (epoch 9), clean test DICE score, and corrupted test DICE score.

Training Setup	Final Val. Loss	Clean DICE	Corrupted DICE
No Augmentation	0.083	0.945	0.512
Train1	0.108	0.929	0.735
Train2	0.082	0.931	0.714
Train3	0.117	0.912	0.652



(a) Example segmentation under the standard model without augmentation

(b) Example segmentation under customized model with augmentation

Figure 5: Qualitative results showing model performance under domain-specific corruptions. Our augmentation improves robustness in surgical scenarios.

5. Conclusion

In this study, we propose specialized augmentation strategy to improve the robustness of surgical tool segmentation models under realistic surgical corruptions using the SegSTRONG-C dataset[7]. Our experiments demonstrates that while a vanilla U-Net[2] achieves strong performance on clean data, it fails to generalize when faced with corrupted test cases under bleeding corruptions. This indicates the vulnerability of models trained solely on idealized data. Training with our augmentation stratergy substantially improved model robustness: corrupted test performance improved from 51% (baseline) to over 70% with augmentation, while maintaining a high DICE score on clean inputs. Among the strategies tested, occlusion-based augmentations proved to be the most critical for enhancing robustness, with photometric transformations providing additional incremental gains. Overall, our findings highlight that rather than relying solely on large datasets or synthetic generation, carefully designed domain-specific augmentations can bridge the gap between clean training conditions and the expected of real surgery. This work thus provides a practical pathway towards building more reliable AI systems for robotic surgery.

References

[1] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*, pages 3431–3440, 2015.

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proc. ECCV*, pages 801–818, 2018.

[4] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proc. CVPR*, pages 2881–2890, 2017.

[5] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proc. CVPR*, pages 6881–6890, 2021.

[6] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Proc. NIPS*, 34:12077–12090, 2021.

DOI: 10.54254/2755-2721/2025.LD30738

[7] Hao Ding, Yuqian Zhang, Tuxun Lu, Ruixing Liang, Hongchao Shu, Lalithkumar Seenivasan, Yonghao Long, Qi Dou, Cong Gao, Yicheng Leng, et al. Segstrong-c: Segmenting surgical tools robustly on non-adversarial generated corruptions–an endovis' 24 challenge. *arXiv preprint arXiv:2407.11906*, 2024.

[8] Emanuele Colleoni, Philip Edwards, and Danail Stoyanov. Synthetic and real inputs for tool segmentation in robotic surgery. In *Proc. MICCAI*, pages 700–710. Springer, 2020.

[9] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571, 2016.

[10] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[12] Hao Ding, Jintan Zhang, Peter Kazanzides, Jie Ying Wu, and Mathias Unberath. Carts: Causality-driven robot tool segmentation from vision and kinematics data. In *International conference on medical image computing and computer-assisted intervention*, pages 387–398. Springer, 2022.

[13] Hao Ding, Jie Ying Wu, Zhaoshuo Li, and Mathias Unberath. Rethinking causality-driven robot tool segmentation with temporal constraints. *International Journal of Computer Assisted Radiology and Surgery*, 18(6):1009–1016, 2023.

[14] Nathan Drenkow, Numair Sani, Ilya Shpitser, and Mathias Unberath. A systematic review of robustness in deep learning for computer vision: Mind the gap? *arXiv preprint:2112.00639*, 2021.

[15] Nathan Drenkow, Chris Ribaudo, and Mathias Unberath. Causality-driven audits of model robustness. *arXiv preprint:2410.23494*, 2024.

[16] Nathan Drenkow, Mitchell Pavlak, Keith Harrigian, Ayah Zirikly, Adarsh Subbaswamy, and Mathias Unberath. Detecting dataset bias in medical ai: A generalized and modality-agnostic auditing framework. *arXiv preprint:2503.09969*, 2025.

[17] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[18] Rahul Parhi and Robert D. Nowak. The role of neural network activation functions. *IEEE Signal Processing Letters*, 27:1779–1783, 2020.

[19] Alex Clark et al. Pillow (pil fork) documentation. *readthedocs*, 2015.

[20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, October 2023.