Surgical Tool Segmentation Using DeepLabv3 with Transfer Learning

Aditi Godbole
Department of Computer Science
University of Stuttgart
Email: aditi.godbole.ag@gmail.com

***Abstract*—Surgical tool segmentation plays an essential role in robotic-assisted surgeries, enabling automation, tracking, and scene understanding. In this work, we implement a semantic segmentation pipeline using DeepLabv3 with a ResNet50 backbone. The model is trained on labeled surgical video data and evaluated on an unseen test set using standard metrics. Our results show a mean IoU of 0.68, Dice score of 0.75, and pixel accuracy of 92.4%, highlighting the model's effectiveness in accurately identifying surg...**

I. Introduction

Accurate semantic segmentation of surgical instruments is critical in computer-assisted interventions. It enhances real-time decision-making and supports robotic operations. While several deep learning architectures have been proposed for segmentation tasks, we select DeepLabv3 with ResNet50 as it balances accuracy and efficiency. DeepLabv3 leverages atrous spatial pyramid pooling (ASPP), which captures multi-scale contextual information, making it suitable for recognizing surgical tools of varying size. This work presents a practical implementation of a surgical tool segmentation pipeline using DeepLabv3, evaluated on real surgical data.

II. Dataset and Preprocessing

We used the SAR-RARP50 dataset [1], which contains RGB videos of robotic-assisted surgeries. Each video is stored in a folder named `video_xx` and includes a `video_left.avi` file and its corresponding pixel-level segmentation masks.

A. Frame Extraction

We extracted frames at 1 FPS from the video files using our `extract_all_frames.py` script, which uses OpenCV's `VideoCapture` module. Extracted frames are saved into a `frames/` subdirectory within each video folder. The script computes frame intervals based on the source video FPS and ensures consistent frame resolution and naming.

B. Dataset Class

The `SurgicalDataset` class (in `dataset.py`) loads the RGB images and masks using `cv2`. It supports both training and validation mode and applies Albumentations-based augmentations during training. Masks are loaded in grayscale and resized using nearest-neighbor interpolation to preserve class integrity.

C. Augmentations

For training, we applied horizontal flips, random rotations, affine transformations, and normalization. Validation data is only normalized. Augmentations are implemented using `albumentations.Compose` with `ToTensorV2()` to convert images to PyTorch tensors.

III. Model Architecture

We use DeepLabv3 with a ResNet50 backbone, as implemented in PyTorch [2]. This architecture is well-suited for medical segmentation due to its strong encoder and ASPP-based decoder [3]. The ResNet backbone, pretrained on ImageNet, enables effective feature reuse. The model outputs a dictionary, from which we extract the `["out"]` tensor for segmentation. The final classifier predicts 10 classes (tools + background).

IV. Training Procedure

Training was conducted using `train.py`, which loads data using the `SurgicalDataset` and prepares a `ConcatDataset` of multiple video folders. Frames and masks are resized to 256x256. We use Adam optimizer (learning rate 1e-4) and cross-entropy loss. Training runs for 5 epochs with batch size 2 on a CUDA-enabled GPU. Best model checkpoints are saved to disk.

V. Evaluation Protocol

Evaluation was done using `evaluation.py` across test videos `video_41` to `video_50`. Each video is evaluated independently. Metrics include:

- **Mean Intersection over Union (mIoU)**
- **Dice Score**
- **Pixel Accuracy**

Predictions are compared against ground truth masks, and results are aggregated per video. Outputs are saved to a CSV file (`evaluation_summary.csv`).

VI. Results

Table I presents the performance across all test videos. Pixel accuracy remains consistently above 91%, showing the model's robustness. Dice and IoU scores reflect successful segmentation of smaller tools.

TABLE I
EVALUATION METRICS ON TEST VIDEOS

Video	Mean IoU	Mean Dice	Pixel Accuracy
video_41	0.6815	0.7249	0.9243
video_42	0.7096	0.7649	0.9133
video_43	0.6347	0.6767	0.8654
video_44	0.7284	0.7751	0.9202
video_45	0.8149	0.8690	0.9527
video_46	0.7258	0.7768	0.9183
video_47	0.7610	0.8197	0.9433
video_48	0.7212	0.7685	0.9317
video_49	0.6640	0.7098	0.8874
video_50	0.7315	0.7826	0.9537

VII. CONCLUSION

We presented a full segmentation pipeline for surgical tool identification using DeepLabv3. The method includes automated frame extraction, dataset construction with augmentations, training, and evaluation. Results demonstrate that even with relatively low resolution and simple preprocessing, high pixel accuracy and strong segmentation performance are achievable.

Future work may include class-specific segmentation, multi-frame temporal refinement, and lightweight architectures for real-time use.

REFERENCES

[1] Y. Zhou and et al., "Sar-rarp50: A surgical action recognition dataset for robot-assisted surgery," 2022, https://www.synapse.org/#!Synapse:syn27618412/wiki/616881.

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[3] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.