Real-Time Video Segmentation for Autonomous Robotic Manipulation

Vakula Venkatesh
Stanford University
vakulav@stanford.edu

Chetan Reddy Narayanaswamy
Stanford University
chetanrn@stanford.edu

Abstract

We propose a real-time semantic segmentation framework for robotic surgical scenes, enabling downstream imitation learning for autonomous robotic assistance. Our system is built on top of a custom-collected dataset from the daVinci Surgical System performing object transfer tasks. We leveraged the Segment Anything Model 2 (SAM2) to obtain high-quality masks and trained a lightweight U-Net architecture on them, achieving near-equivalent segmentation performance with 30Hz inference speed, suitable for closed-loop robotic control.

The segmentation masks are intended as inputs to an imitation learning policy for autonomous manipulation with the third surgical arm. We present quantitative comparisons of segmentation quality, model latency, and qualitative outputs across different methods, highlighting our U-Net's balance between performance and efficiency. This work contributes a deployable perception module tailored for surgical robotics and paves the way toward real-time learning-based automation in high-stakes environments. Project Github Repo: https://bit.ly/3SCxd5y

1. Introduction

1.1. Motivation & Problem Statement

In robot-assisted surgery, precise real-time perception is essential for enabling closed-loop control and decision-making. The da Vinci Surgical System, while powerful, currently operates under full teleoperation, limiting its scalability in tasks requiring coordination across multiple robotic arms. Automating the third arm can significantly enhance surgical dexterity, but this requires robust and efficient perception of the scene [6], particularly, the ability to segment robot arms and manipulated objects accurately in real time.

Existing zero-shot segmentation methods like SAM2 offer high-quality masks but are too slow for real-time robotic control, and text prompt-based segmentation using CLIPSeg produce coarse outputs and struggle with surgical scene complexity. Classical computer vision techniques are fast but fail to generalize due to occlusions, lighting variation, and visual similarity between tools and background.

Our objective is to build a segmentation module that operates at 30Hz, producing semantic masks from surgical video frames that can serve as inputs to an imitation learning policy. This module must maintain segmentation quality comparable to SAM2 while achieving real-time inference speeds suitable for robotic deployment.

1.2. Project Scope & Goals

The input to our system is a 30Hz RGB video stream from surgical demonstrations. The output is a binary segmentation mask for each frame that highlights the robot arms and the manipulated object. These masks shall serve as input features for training an imitation learning policy that aims to automate the third surgical arm, enabling it to act collaboratively with the two surgeon-controlled arms, which will benefit from the real-time segmentation in the future.

To achieve this, we explored and benchmarked multiple segmentation approaches, including classical computer vision methods, prompt-based segmentation using CLIPSeg, foundation model-based segmentation with SAM2 and video propagation, and a lightweight U-Net trained on SAM2-generated masks for fast inference.

Our ultimate goal is to bridge the gap between high-quality but slow segmentation, and real-time deployment needs, delivering a segmentation pipeline that enables low-latency perception for closed-loop robotic control.

2. Related Work

2.1. Foundation Models for Segmentation

Recent advancements in foundation models have significantly improved segmentation performance across diverse domains. The Segment Anything Model (SAM) and its successor SAM2 represent a shift toward promptable, general-purpose segmentation using large-scale pretraining. SAM2, in particular, extends segmentation to the video domain by incorporating a video propagation module that uses cross-frame attention to generate consistent masks across frames [7].

While SAM2 provides high-quality masks with minimal supervision, its inference speed is a major limitation. Full video propagation on short surgical sequences can take several minutes, making it infeasible for real-time robotic applications. Additionally, SAM2 requires manual initialization via prompts such as points or bounding boxes, which adds human-in-the-loop latency.

To address domain adaptation challenges, MedSAM and MedSAM2 have fine-tuned these models specifically on medical imaging data. MedSAM2 adapts SAM2's video segmentation pipeline for surgical scenes, offering better robustness to domain-specific noise and structure [10]. However, these models are still large, often require GPU acceleration for inference, and do not satisfy the low-latency constraints of robotic control.

In this work, we use SAM2's high-quality outputs as pseudo-labels to train a smaller U-Net model capable of running in real time, thus combining the benefits of foundation model supervision with lightweight deployment.

2.2. Robotic Manipulation Scene Segmentation

Accurate segmentation allows robots to perceive their environment with spatial precision and supports higher-level autonomy in complex tasks. Among deep learning architectures, U-Net has become a widely used model for dense prediction tasks due to its encoder-decoder structure and skip connections that preserve fine-grained details. Variants such as U-Net++ and Attention U-Net have improved performance in limited-data regimes and structured environments [8].

In the domain of robotic manipulation, segmentation models often face challenges such as occlusions, background clutter, dynamic lighting, and tool-object interactions. While datasets like EndoVis [1] have driven progress in segmentation for surgical tools, similar challenges persist across manipulation domains where tools, hands, or arms may blend visually with the background.

Traditional segmentation models must often be trained from scratch for each environment or task, which limits scalability and deployment. To overcome this, our work leverages pseudo-labels generated by a powerful foundation model (SAM2) to train a compact U-Net, enabling real-time and generalizable segmentation in robotic manipulation scenes.

2.3. Real-time Semantic Segmentation

In robotics applications, including manipulation and control, real-time perception is essential to ensure low-latency feedback and safe operation. This has motivated the development of lightweight segmentation models optimized for both speed and accuracy.

ENet [4] introduced one of the earliest real-time segmentation networks, achieving high inference speed with significantly fewer parameters than traditional models like DeepLab. BiSeNet [9] improves on this by decoupling spatial and contextual feature extraction, enabling high-resolution output at low latency. Fast-SCNN [5] follows a similar philosophy by using a two-branch encoder and a lightweight decoder for mobile and embedded use cases.

Although such models are appealing for their speed and low hardware demands, they often trade off segmentation accuracy for speed. Our approach builds upon this insight by training a compact U-Net on high-quality labels generated by SAM2, thereby retaining accuracy while achieving real-time performance.

3. Dataset

We collected the data using the daVinci Surgical Robot at the Stanford Robotics Center for two manipulation tasks. The details about the two datasets are given in table 1. The ultimate objective of the imitation learning objective is to achieve collaboration between the human controlled arms and and the AI controlled arms. Therefore, in the data collection phase, two humans are included who collaboratively manipulate the arms. The tasks are described below.

- **Object Transfer Task:** One arm (controlled by person A) picks up an object and passes it to a second arm (controlled by person B), which must receive it accordingly.
- **Object Shifting Task**: Three arms synchronously pick up three objects and place them into a bowl. Two arms are controlled by person A, and the third arm is controlled by person B.

3.1. Processing the Video Data

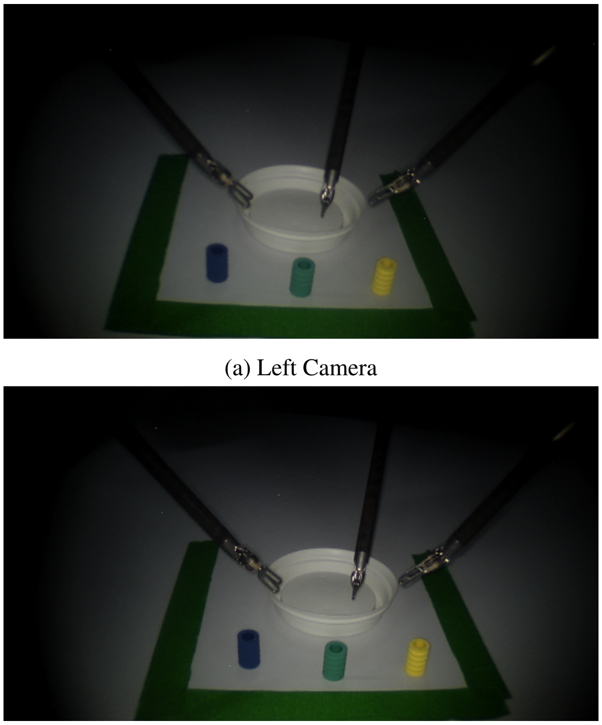
Both the kinematic and vision data are recorded at 30 Hz but we focus on the vision data for this work. The resolution of the recorded RGB frames is 576x324. The video data is available from two cameras as shown in figure 1. For the UNet model, the resolution is reduced to

Table 1: Details about the Tasks in the Dataset

Properties	**Object Transfer**	**Object Shifting**
Number of Robotic Arms	2	3
Frequency of Collected Datapoints	30 Hz	30 Hz
Duration per Demonstration	15 s	20 s
Number of Frames per Demonstration	450	600
Number of Demonstrations	100	70

256x256 while the original resolution is retained for the SAM2 model. Further, we resampled the recorded videos to 10 Hz reducing the number of frames per demonstration from 600 to 200.

We did not perform extensive data augmentation, as our goal is to develop a lightweight model tailored specifically to the tasks described above. The combined number of frames from all demonstrations is sufficiently large to train the model effectively without overfitting.



(a) Left Camera

(b) Right Camera

Figure 1: Initial Video Frame for the Object Shifting Task

4. Methods

4.1. Pipeline

To build a lightweight real-time segmentation module for surgical manipulation scenes, which can operate at 30Hz and serve as a perception front-end for an imitation learning policy, we designed a two-stage pipeline that leverages the segmentation capabilities of SAM2, and the speed and efficiency of a U-Net for deployment. This is depicted in Figure

The input to our pipeline is a video stream from the scene recorded using the camera on the robot. We use SAM2 in the offline phase to generate the segmentation masks for surgical arms and the manipulation object across each video. These masks then serve as pseudo-ground-truth labels to train a compact U-Net architecture on downsampled 256×256 RGB frames.

Once trained, the U-Net model is capable of segmenting new video frames in real time, achieving inference speeds of up to 30Hz.

4.2. Baseline Methods

Before settling on a learning-based segmentation approach, we explored two baseline methods: a classical computer vision pipeline and a zero-shot prompt-based segmentation model (CLIPSeg). These served both as sanity checks and to understand the performance gap between traditional techniques and modern vision models.

4.2.1 Classical CV Baseline:

We attempted segmentation using HSV thresholding to detect the object and robotic arms. This was a largely unsuccessful attempt, as the arms often blended with the environment due to similar texture and color profiles, and the masks were noisy and inconsistent across frames (as seen in 4. Additionally, the approach lacked temporal consistency, making it unsuitable for downstream tasks like video segmentation or control.

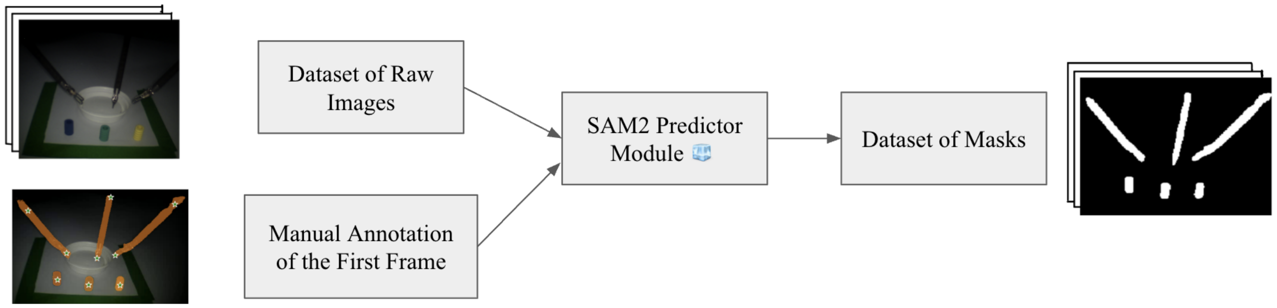


Figure 2: SAM2 Pipeline

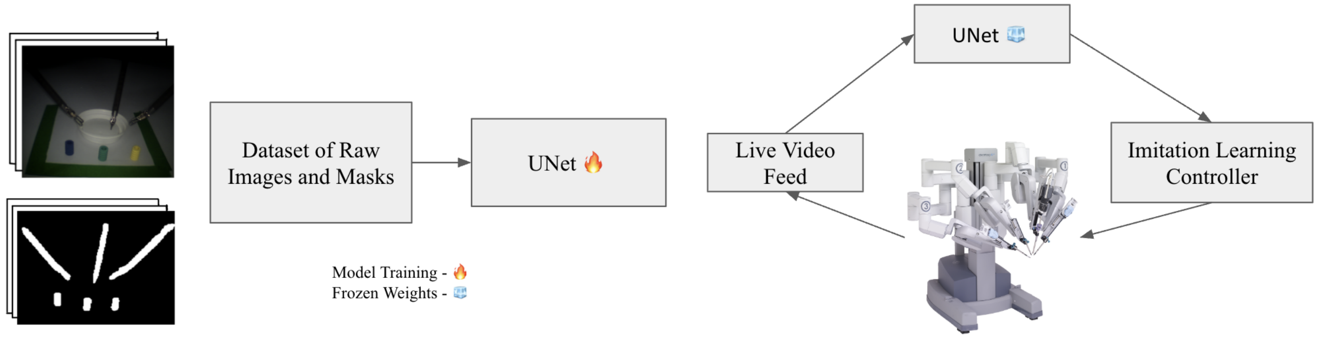


Figure 3: U-Net Training and the broader Imitation Learning Pipeline

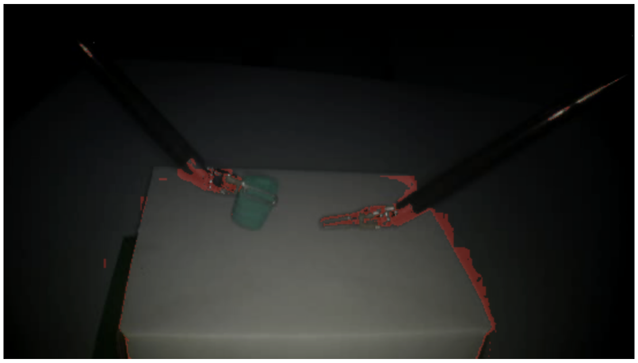


Figure 4: Classical CV Masking Output

4.2.2 CLIPSeg:

We also evaluated CLIPSeg, that performs zero-shot segmentation based on natural language prompts [3]. While CLIPSeg provided some semantically meaningful outputs, they resulted in coarser masks in comparison to the masked outputs from SAM2. Furthermore, prompt tuning was not reliable in surgical scenes where classes like ”`robotic arm`” or ”`small cylindrical object`” are underrepresented in CLIPSeg's training corpus, the inconsistency of which can be seen in fig 5.

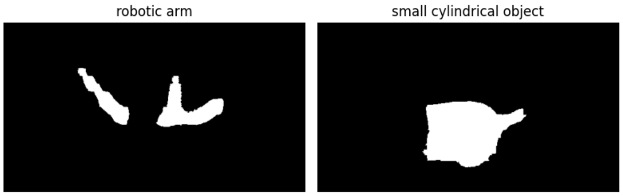


Figure 5: CLIPSeg Output

4.3. U-Net Architecture & Real-Time Inference

We implemented a lightweight U-Net architecture tailored for binary mask prediction of robot arms and objects in the scene. The architecture follows the standard U-Net encoder-decoder design with skip connections to preserve spatial information across layers.

4.3.1 Architecture Design

The encoder comprises two convolutional blocks, each containing two sequential 3×3 convolutions followed by ReLU activations. MaxPooling is applied after each block to downsample spatial dimensions. The bottleneck layer contains two 3×3 convolutions with 256 channels. The decoder mirrors the encoder with two transposed convolutions for upsampling, each followed by concatenation with the corresponding encoder features and a pair of 3×3 convolutions. A final 1×1 convolution projects the output to a single-channel binary mask, and a sigmoid activation converts logits into probabilities. All input images are resized to 256×256 for training and inference. An exponent of 2 was chosen for consistent halved downsampling over multiple blocks.

4.3.2 Real-Time Segmentation Deployment

We deployed the trained model that accepts the live feed from the camera mounted on the bot. Each frame is resized and passed through the trained U-Net, which produces a binary mask in under 30 ms per frame, achieving ∼30Hz throughput on GPU. The segmentation mask is postprocessed to match the original input resolution and is displayed side-by-side with the raw video in real time.

5. Experiments & Results

5.1. Pseudo-Ground-Truth Mask Generation using SAM2

The mask generation was carried out in two stages - manual annotation on the raw-images obtained from the bot to provide the region of interests that the model must segment, and propagation of these masks across the video to generate masks for 600 frames of a 20s video object shifting task (as shown by initial video frames in 1). The model employs cross-frame memory attention to iteratively refine masks and maintain consistency across frames.

A predictor module is first built which loads the SAM2 model using the corresponding model configuration and checkpoint. Here, we have used a small model configuration (`sam2.1_hiera_small.pt`) to understand propagation time for a 20s video. We then initialize the predictor and obtain the inference state, and run it over all our frames. This initialization process takes ∼30s on a GPU.

We then register clicks on one frame from the video and obtain the mask logits by providing the selected region to the `add_new_points_or_box` module of the model, as can be seen in 6.

The inference state is then passed to the `propagate_in_video` module of the predictor to obtain mask logits for all the frames. This produces segmented binary masks (as shown in 9a) after a computation time of ∼23 minutes on a GPU.

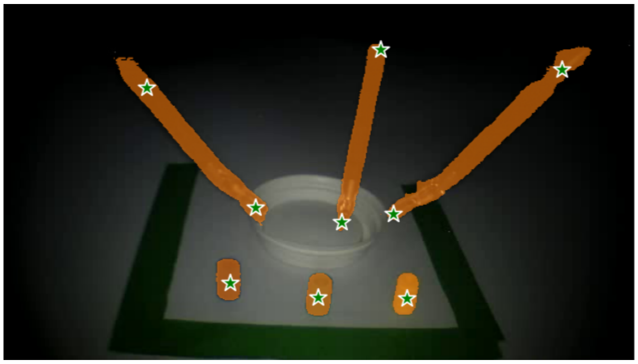


Figure 6: Annotated input to SAM2 for segmentation

5.2. Training Procedure

The model is trained on the binary masks generated by SAM2. Input frames are normalized, resized, and Binary Cross-Entropy (BCE) loss is used for training, optimized using Adam. Training was performed on a single GPU (RTX 4060). The training runs were logged on Weights and Biases [2] for efficient hyperparameter tuning and visualizations.

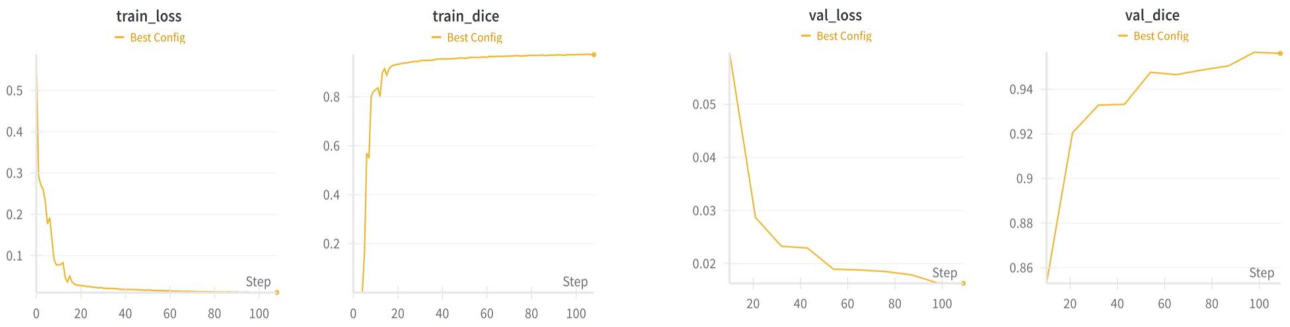
The model demonstrated reliable convergence, with training and validation Dice scores stabilizing after roughly 60 epochs. Visual inspection of predicted masks showed that the UNet model was able to accurately segment object boundaries.

We performed a series of hyperparameter tuning experiments to balance training stability and segmentation performance. We experimented with batch sizes of 2, 4, and 8 and found that a batch size of 4 offered the best trade-off between convergence speed and generalization. The learning rate was tuned manually from 1e-3 to 1e-5, with 1e-4 yielding the most stable training without gradient explosion or vanishing. Dice score was chosen as the main evaluation metric due to the class imbalance in the foreground-background segmentation task. While the model was trained using a default classification threshold of 0.5, further tuning of this threshold during inference may help prioritize high-precision predictions, especially when false positives could lead to incorrect robotic manipulation behavior.

5.3. Evaluation Metrics

To assess the segmentation models quantitatively, we use two primary metrics:

- **Inference Time:**
  It is measured in milliseconds and captures the processing time of the model during a forward pass. It is a critical metric for the evaluation of our task-specific



(a) Training Loss and Training Dice Score

(b) Validation Loss and Validation Dice Score

Figure 7: Training and validation metrics over epochs.

outputs, where frames must be processed at or above 30 Hz.

- **Dice Score:**
  We use the Dice coefficient as a primary metric for evaluating segmentation performance. It is a statistical measure used to evaluate the similarity between two binary masks, commonly used in image segmentation tasks.

$$\text{Dice} = \frac{2|P \cap G|}{|P| + |G|} \tag{1}$$

  - P: predicted binary mask
  - G: ground truth mask
  - $|P \cap G|$: number of overlapping foreground pixels
  - $|P|$, $|G|$: number of foreground pixels in the prediction and ground truth, respectively.

5.4. Quantitative Measures

5.4.1 Inference Time Comparison

As seen in Table 2, although SAM2 produces high-quality masks, it has a runtime of approximately 600 ms per frame, limiting its throughput to around 1.6 frames per second. In contrast, our U-Net model achieves 80 ms per frame on CPU (12.5 FPS) and 10 ms per frame on GPU (100 FPS), demonstrating a 60× speedup over SAM2 when deployed on GPU. This substantial runtime advantage enables our model to operate comfortably at the target rate of 30Hz for real-time robotic control.

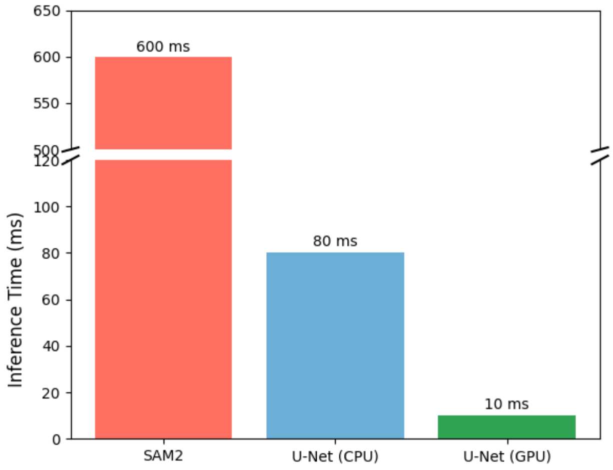


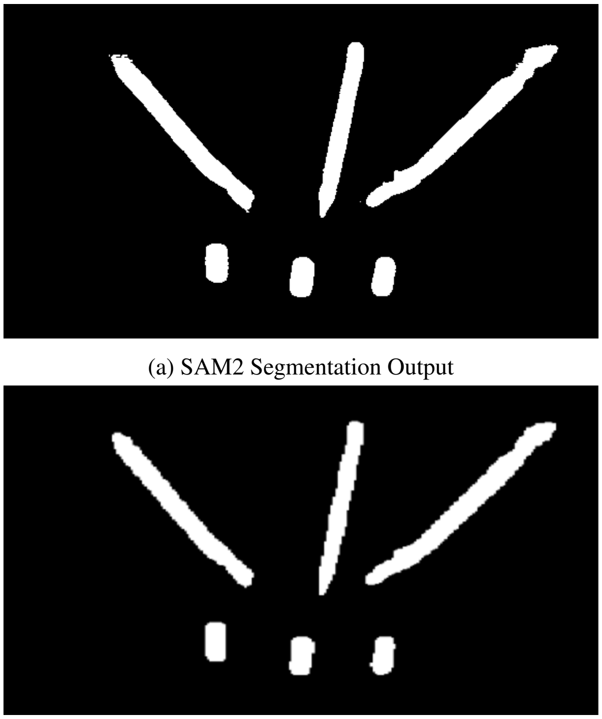
Figure 8: Inference Time Comparison

Table 2: Segmentation performance and runtime comparison across models

Model	**Dice (%)**	**Inference Time (ms)**	**FPS**
SAM2	100	600	1.6
U-Net (GPU)	**95.6**	**10**	**100**
U-Net (CPU)	95.6	80	12.5
CLIPSeg	81.3	150	6.6
Classical CV	63.2	5	200

5.4.2 Dice Coefficient Comparison

As shown in Table 2, SAM2 has been assigned at 100%, as it serves as the pseudo-ground-truth.



(a) SAM2 Segmentation Output

(b) U-Net Segmentation Output

Figure 9: Qualitative comparison of segmentation outputs from SAM2 and U-Net.

5.5. Qualitative Outputs

The output of the U-Net based segmentation is shown in fig. 9b, alongside the output of SAM2 segmentation. It can be seen that the segmentation output of U-Net is identical to that of SAM2, which is also verified by the quantitative measures above.

The obtained video output of the U-Net also proved to be satisfactorily well to run in real-time alongside the live camera feed. *The video outputs can be viewed on the project GitHub Repository* [1].

6. Conclusion & Future Work

In this work, we presented a real-time semantic segmentation pipeline tailored to our use case employing the daVinci Surgical Robot, leveraging the strengths of foundation models to generate supervised data while ensuring low-latency inference for deployment. By using SAM2 to generate segmentation masks, we avoided the need for costly manual annotation. These masks were then used to train a lightweight U-Net, enabling accurate and fast mask prediction at 30Hz, which is a critical requirement for closed-loop control in robotic manipulation in surgical scenarios. Our approach demonstrated that task-specific real-time models can closely approximate the performance of large-scale segmentation models when trained with high-quality supervision, without incurring the runtime costs of those models.

Our future work is targeted at integrating the segmentation output into an imitation learning framework, where the masks shall serve as inputs to policy networks controlling the third robotic arm. Additionally, we plan to explore the estimation of depth from our RGB input sequences, in order to provide more specified data for the imitation learning framework.

From a sequence of video frames, it is also possible to predict the specific subtask being performed. This information can be valuable for enabling the automated arm to assist the teleoperated arms more intelligently. The problem can be framed as a video classification task with predefined labels, which is also something we plan to execute next.

7. Contribution & Acknowledgments

This project was a joint effort. The SAM2-based segmentation experiments and prompt-driven CLIPSeg evaluations were conducted by Vakula Venkatesh. The U-Net segmentation pipeline and classical computer vision baselines were developed and implemented by Chetan Reddy. Both team members were equally involved in dataset collection using the da Vinci Surgical System, as well as in formulating the overall project structure, designing the segmentation pipeline, and evaluating the system's performance.

We are grateful to our project mentor Rahul Venkatesh for his continued and prompt guidance and for providing us with valuable inputs to validate and refine our pipeline. We also acknowledge the CS231N teaching team for their guidance throughout the project.

[1] https://github.com/chetanreddyn/Video-Segmentation-for-Autonomous-Manipulation.git

References

[1] M. Allan, S. Kondo, S. Bodenstedt, S. Leger, R. Kadkhodamohammadi, I. Luengo, F. Fuentes, E. Flouty, A. Mohammed, M. Pedersen, et al. 2018 robotic scene segmentation challenge. *arXiv preprint arXiv:2001.11190*, 2020. 2

[2] L. Biewald. Experiment tracking with weights and biases. `https://www.wandb.com/`, 2020. 5

[3] T. Lüddecke and A. Ecker. Image segmentation using text and image prompts. pages 7086–7096, 2022. 4

[4] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016. 2

[5] R. P. Poudel, S. Liwicki, and R. Cipolla. Fast-scnn: Fast semantic segmentation network. *arXiv preprint arXiv:1902.04502*, 2019. 2

[6] L. Qiu, C. Li, and H. Ren. Real-time surgical instrument tracking in robot-assisted surgery using multi-domain convolutional neural network. *Healthcare technology letters*, 6(6):159–164, 2019. 1

[7] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2

[8] N. Siddique, P. Sidike, C. Elkin, and V. Devabhaktuni. U-net and its variants for medical image segmentation: theory and applications. *arXiv preprint arXiv:2011.01118*, 2020. 2

[9] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. pages 325–341, 2018. 2

[10] J. Zhu, A. Hamdi, Y. Qi, Y. Jin, and J. Wu. Medical sam 2: Segment medical images as video via segment anything model 2. *arXiv preprint arXiv:2408.00874*, 2024. 2