Vision-based Surgical Instrument Segmentation and Endoscopic Sinus Surgery Skill Assessment

Shan Lin

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2021

Reading Committee:
Blake Hannaford, Chair
Jenq-Neng Hwang
Samuel Burden

Program Authorized to Offer Degree:
Electrical and Computer Engineering

University of Washington

Abstract

Vision-based Surgical Instrument Segmentation and
Endoscopic Sinus Surgery Skill Assessment

Shan Lin

Chair of the Supervisory Committee:
Professor Blake Hannaford
Electrical and Computer Engineering

In robot-assisted surgery, engineering technologies are applied to augment surgeons' ability to conduct safer surgeries and achieve better treatment outcomes. To provide appropriate assistance to the surgeons, the ability to understand surgical phases, and predict risks and remaining procedures is an enabling technology for next-generation autonomous surgical robots. Vision-based surgical instrument segmentation, which aims to identify instrument regions in surgical images, is one important task that can provide instrument location information to robotic systems. The potential uses of instrument segmentation results include surgical workflow analysis, risk supervision and prediction, and surgical skill assessment. Despite the wide range of potential applications of instrument segmentation, existing technologies are still not robust enough and lack generalizability when applied to highly dynamic surgical environments. In this dissertation, we develop a convolutional neural network that can aggregate video frame features temporally in a recurrent mode to achieve robust segmentation. Moreover, we explore transfer learning technologies that can improve segmentation performance on the target domains by leveraging knowledge learned from labeled source domains. Finally, we conduct several pilot experiments on vision-based, automatic and objective surgical skill assessment.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my PhD advisor Professor Blake Hannaford, for his everlasting support and enlightening guidance. I learned a lot from him, not only domain-specific knowledge and envision, but also how to conduct innovative research. He also showed me a good example of how to guide students to explore the academic world. I am also really thankful for the wonderful academic connections and resources he introduced to me during my job search in academia. Without his support, I could not imagine how I could have achieved this so far. Moreover, I would like to thank my Master's advisors Professor Robert J. Webster III at Vanderbilt University and Professor Loris Fichera at Worcester Polytechnic Institute, for their awesome advice throughout my master's years, which helped me to establish my career goals, and encouragement and career suggestions post my master's.

Next, I would like to thank the rest of my PhD committee: Professor Jenq-Neng Hwang, Professor Samuel Burden, and Professor Eric Seibel, for their continuous and insightful advice and encouragement. I would also like to thank Professor Randall A. Bly and Professor Kris S. Moe, for collecting the medical data to support my continuous research and offering professional advice from the medical side. Furthermore, I really appreciate the productive collaborations and discussions with my former and current lab mates Dr. Fangbo Qin, Professor Yangming Li at Rochester Institute of Technology, Yun-Hsuan (Melody) Su at Mount Holyoke College, Niveditha Kalavakonda, Haonan Peng, and Andrew Lewis, which resulted in numerous achievements that I cannot accomplish by myself.

Last but not least, I would like to thank my parents Jinbiao Lin and Bing Lu, for their unconditional support and selfless love, as well as valuable career advice. They always believe

that I can finally accomplish my goals and support my decisions. And special thanks to my fiancé Huazeng Deng, for his forever support during my PhD and especially, the amazing food he cooked which made me gain $+\infty$ pounds in weight.

DEDICATION

To my parents, Jinbiao Lin and Bing Lu.

Chapter 1
INTRODUCTION

Over the past few decades, minimally invasive surgery and robotic-assisted surgery have been revolutionizing the delivery of health care. These advanced technologies could bring many benefits to patients, including smaller incisions, less pain and bleeding, better clinical outcomes, and a shorter hospital stay [10, 83]. However, they also introduce challenges to the surgeons. For example, in endoscopy, a tiny camera called the endoscope is inserted into the body to provide a real-time view of the surgical site. The endoscope provides limited field-of-view and reduced depth perception that impacts the surgeons' eye-hand coordination ability [9, 10]. Additionally, a limited sense of touch further reduces the information a surgeon can get from the surgical site [107, 103]. Therefore, systems or robots that can understand the current surgical status and provide appropriate assistance are highly needed.

Artificial intelligence (AI) and machine learning (ML) have been widely explored for various tasks in healthcare. There are still many unsolved problems that impede the application of AI and ML to medical problems. More specifically, in this dissertation, we focus on vision-based surgical instrument segmentation, an important task for computer and robotic-assisted surgery, which aims to outline the instrument in surgical images. Despite the advanced segmentation results obtained with state-of-the-art models, generalization ability is still limited. Further, these models are not robust enough to highly dynamic surgical environments.

1.1 Vision-based Surgical Instrument Segmentation

1.1.1 Motivations

Surgical instrument segmentation is an important component of computer or robotic-assisted surgery. Figure 1.1 shows some potential applications of instrument segmentation. By iden-

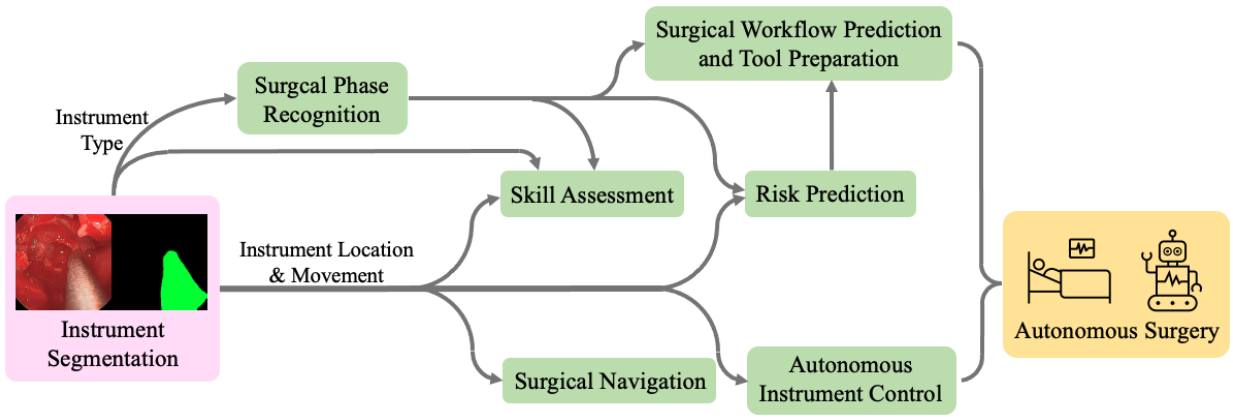

Figure 1.1: Potential applications of surgical instrument segmentation. Note that some tasks may require information besides instrument segmentation results.

tifying instruments through surgical videos, relative movements between the instrument and the endoscope can be extracted. Besides vision-based methods, optical, ultrasonic, electromagnetic, and RFID systems have been explored and embedded in some surgical procedures to track surgical instruments [10]. However, these systems are generally expensive, bulky, and may restrict the surgeons' movements [100]. Some tracking data can be missing due to inherent technology limits, for example, optical tracking systems require the markers to be in the light-of-sight of the camera [100]. In contrast, vision-based methods are non-invasive and do not require changes to current surgical workflows since videos are already well integrated with various surgical procedures. Also, since surgeons rely on videos to perform surgeries, the instruments are almost always visible in videos during operations. Therefore, vision-based instrument segmentation and tacking can be complementary or alternation to other tracking systems. For robotic surgery, the vision-based segmentation results could be combined with kinematic data to achieve more accurate tracking [4, 96]. On the other hand, instrument type is another information that could be extracted through instrument segmentation. While our current focus is image region segmentation, we will explore methods to identify instrument types in the future.

To provide appropriate assistance to surgeons or perform surgical tasks autonomously,

intelligent systems or robots should be able to understand surgery procedures, be aware of surgeons' operation quality, as well as identify potential risks. Knowing the relative location between instruments and important anatomical structures could allow the system to guide the surgeons, detect potential risks and inform surgeons, and allow robots to autonomously manipulate the instrument accurately. Moreover, both the instrument and background regions in the videos reveal the surgical skill level: the instrument regions in the videos are related to the relative movements between the instrument and the camera (e.g., endoscope), while the background reveals the surgical procedure and the movements of the camera. The instrument segmentation results could be used to guide advanced algorithms to identify different clues related to skill levels. Further, the use of certain instrument types indicates the surgical skill level and is related to different surgical phases [58].

1.1.2 Existing work and Challenges

According to the review paper [10], methods to detect instruments can be roughly classified into two types, marker-based and marker-less. Marker-based methods use markers attached to instruments to assist detection. Common marker-based methods include optical, ultrasound, electromagnetic, and RFID tracking [10, 100], which have been discussed in Section seg-motivation. In this work, we focus on marker-less instrument segmentation methods which require no modifications of existing surgical instruments. Traditional machine learning methods such as random forest have been used based on hand-crafted features including color, texture, and gradient information [7, 10, 11]. Since FCN and U-Net were proposed in 2015 [81, 102], the encoder-decoder architecture with skip connections to fuse the features extracted by the encoder with the decoder (see Figure 1.2) has achieved cutting-edge segmentation results on the surgical instrument segmentation task [39, 17, 53, 18, 106, 55, 90, 97, 3, 2]. Researchers have mod-

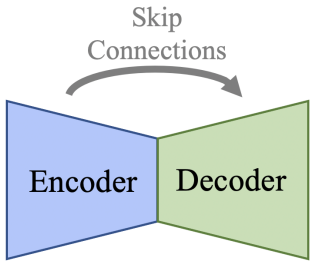

Figure 1.2: Schematic of the encoder-decoder architecture with skip connections.

ified network architectures to obtain better feature representation by extracting and fusing multi-scale context information [18, 77, 55, 97]. Training the segmentation network with multiple related tasks at the same time has also been explored [55, 97, 39]. More recently, the attention mechanisms that can guide the network to extract more relevant information become more appealing [90, 91, 34].

Despite these efforts, existing technologies are still not robust enough to the presence of challenging conditions in surgical images such as varying illumination, specular reflection and blood (see details in Section 2). Moreover, due to highly dynamic surgical environments, there is a large visual appearance diversity in images of different types of surgery and images collected from different hospitals. Existing segmentation models usually do not generalize well to different datasets, and labeling a large amount of training data is needed to improve performance when dealing with a new dataset. Generalization ability is not a problem specific to surgical instrument segmentation. Instead, transfer learning is a type of machine learning that aims to improve performance on new tasks by leveraging knowledge learned from related tasks. Transfer learning has attracted increasing attention for a wide range of tasks. In this work, we explore a sub-field of transfer learning called domain adaptation to transfer knowledge of the instrument segmentation task between different surgical datasets. Also, since algorithm failure may lead to serious consequences in medical applications, improving model generalization ability is essential.

1.2 Automatic and Objective Surgical Skill Assessment

1.2.1 Motivations

Currently, medical students are still trained under an apprenticeship model that has been used for more than a century [16]. The students' surgical skill level is rated by senior surgeons, which is subjective and time-consuming [100, 1]. To standardize the evaluation process, rating rubrics such as Objective Structured Assessment of Technical Skill (OSATS) and Objective Structured Clinical Examination (OSCE) have been proposed [1, 66, 86].

Students are evaluated with the scores of skill-related metrics in the rubrics, but this rating method is still tedious and has a certain degree of subjectivity. Although increasing new surgical technologies require medical students to acquire more skills, the existing evaluation approaches are not efficient enough. To address this dilemma, automated and objective surgical skill assessment methods are highly needed [40, 58].

The ability to analyze surgical skills will also benefit the development of surgical robots that have a higher autonomy level, as briefly described in Section 1.1.1. Accurate skill assessment could allow robots to provide more appropriate information and assistance to surgeons. Also, surgical skill information has the potential to boost advanced machine learning technologies, such as reinforcement learning and imitation learning [50], that have shown promising results in enabling robots to perform complex tasks. For instance, the skill information could be considered when calculating feedbacks in reinforcement learning algorithms for autonomous surgical robots. For imitation learning, the skill levels could be used to rank demonstrations and potentially allow for performance better than demonstrations [13].

1.2.2 Existing work and Challenges

Most previous objective surgical skill assessment methods focus on extracting motion metrics such as average velocity and movement smoothness from the instrument's movement trajectories in the 3D space or videos [93, 47, 70, 37, 40]. The extracted metrics are then used as features for skill level classifiers, or are used to conduct statistical analysis discriminating between surgical operations of different skill levels. Although promising results have been achieved, there exist several problems that have not been addressed by these methods. Most datasets explored in previous studies are based on dry labs [38], while skill evaluation methods on real surgeries are also highly needed. As collecting such datasets is very costly and time-consuming, the number of public datasets are limited, constraining researchers' work and the development of this area. In this work, we develop and release a cadaveric endoscopic sinus surgery dataset with skill levels labeled by three experts based on OSATS to explore skill assessment on real surgeries (see Section 2.1.2). On the other hand, analyzing

instrument movements without knowing how surgeons identify or modify certain anatomical structures is far from enough [66]. Recently, there are several studies analyzed videos for skill-related information besides instrument movement [29, 6, 111], but further studies are needed.

1.3 Proposed Study

The contributions of this dissertation are:

- Develop an algorithm that leverages temporal information in surgical videos to achieve robust instrument segmentation.
- Develop domain adaptation methods that allow models to be more generalizable.
- Conduct initial studies on objective surgical skill assessment for endoscopic sinus surgery based on the segmentation results.

The rest of this dissertation is outlined as follows: Chapter 2 describes an endoscopic surgical dataset named UW-Sinus-Surgery-C/L proposed in this work. In Chapter 3, we propose a model that can achieve more robust segmentation by leveraging temporal information between video frames. In Chapter 4, two domain adaptation works that aim to translate model knowledge between different surgical datasets are presented. Next, some initial studies on objective surgical skill assessment are presented in Chapter 5. Finally, a summary of current results and discussions of future work are presented in Chapter 6.

Chapter 2

DATASETS

2.1 Endoscopic Sinus Surgery Dataset: UW-Sinus-Surgery-C/L

Endoscopic sinus surgery is a surgical procedure of removing blockages in the natural drainage pathways of the sinuses to prevent infection and restore their function [61]. In this operation, the surgeon uses one hand to insert a tiny camera called endoscope into the nose to get a real-time view of the surgical site, and use another hand to control the surgical instruments. The endoscope provides limited field-of-view and reduced depth perception that impacts the surgeons' eye-hand coordination ability [9, 10]. Limited sense of touch further reduces the information a surgeon can get from the surgical site [107, 103]. Additionally, sinuses are located around many important anatomical structures such as brain and eyes, making this operation more challenging [67]. Therefore, there is a high revision rate of about 20% often due to incomplete surgery [15, 62]. Considering over 350,000 endoscopic sinus surgeries are implemented in the United States annually [88], publicly available datasets are highly needed for developing technologies that can assist surgeons to achieve better treatment outcomes.

2.1.1 Data Collection

In this work, we published a endoscopic sinus surgery dataset named UW-Sinus-Surgery-C/L, which consists of a cadaver and a live endoscopic sinus surgery dataset [76].[1] The benefits of collecting the cadaver dataset besides the live dataset include: 1) many medical training or study processes are conducted on cadavers, and 2) collecting data from cadavers allows more flexible experiment settings. The cadaver dataset consists of 10, 5-23 minute cadaveric

[1]The dataset is available at `https://digital.lib.washington.edu/researchworks/handle/1773/45396`.

surgery videos with a resolution of 320×240. The live dataset consists of 3, 12-66 minute live surgery videos with a resolution of 1920×1080. The surgeries were recorded at Harborview Medical Center using a Stryker 1088 HD camera system and the Karl Storz Hopkins Ø4mm 0° endoscope. Besides, the location of the instrument and endoscope during the cadaverc surgeries were recorded by the Medtronic Stealth Station S7 surgical navigation system.

2.1.2 Data Labeling

A total of 4345 frames were extracted with a sampling rate of 0.5 Hz from the cadaveric videos. Similarly, a total of 4658 frames were extracted with a sampling rate of 1 Hz from the live videos. Endoscopic frames have large black border regions that contain no useful information, so we extracted the frame center regions and downscaled them to 240×240. Some sample video frames are presented in Figure 2.1. The instrument regions were manually labeled for the segmentation task.

To reduce bias caused by data split and evaluate the generalization ability of the proposed models, we provided a 3-fold cross-validation setup: (1) Each fold of the cadaver dataset consists of 3 to 4 videos. Specifically, the 10 videos in the cadaver dataset are split into the following 3 folds: 4 videos (Procedure ID: 1, 2, 3, and 4) with 1323 labeled frames, 3 videos (ID: 5, 6, and 7) with 1585 labeled frames, and 3 videos (ID: 8, 9, and 10) with 1437 labeled frames. (2) For the live dataset, the 3 folds are the 1st video with 1154 labeled frames, the 2nd video with 2801 labeled frames, and the 3rd video with 703 labeled frames.

In the cadaver dataset, 5 videos were performed by senior surgeons and the remaining 5 videos were from residents. To further assist surgical skill assessment, the videos were labeled by 3 experts based on a Objective Structured Assessment of Technical Skill (OSATS) modified for rating surgeons' performance in endoscopic sinus surgery [66]. The OSATS consists of two groups of metrics: task specific checklist, and global rating. Each metric is scored on a 1 to 5 scale and a higher score means better performance. The task specific checklist focuses on skills presented in different steps during the surgery, and global rating evaluates cognitive and procedural skills throughout the surgery. Because the surgical steps

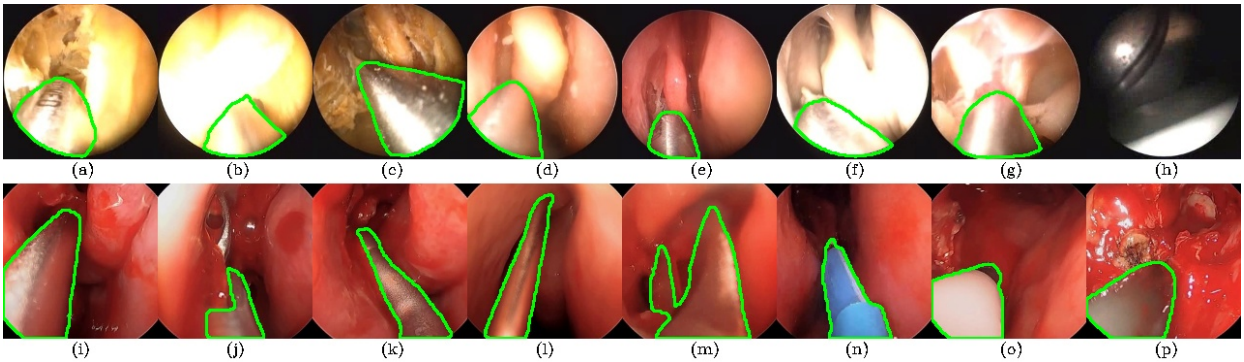

Figure 2.1: Examples of video frames in UW-Sinus-Surgery-C/L. The top row is from the cadaveric surgery videos and the bottom row is from the live surgery videos. The instrument contours are marked with green lines.

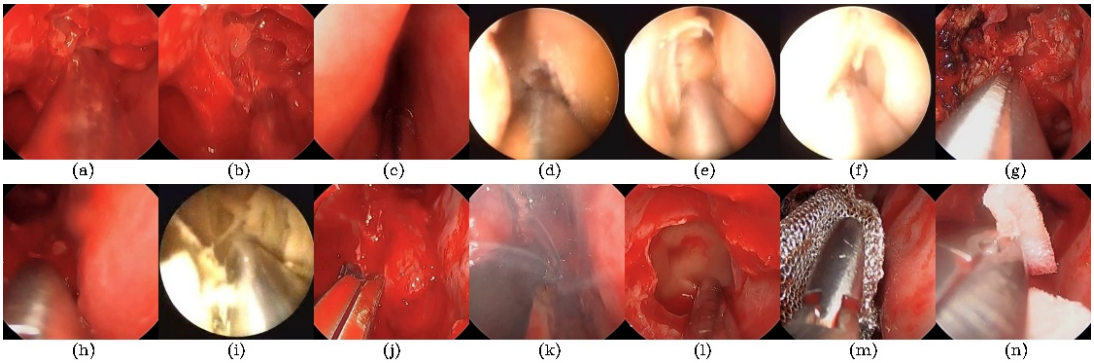

Figure 2.2: Examples of challenging video frames. Challenging conditions: (a-e) background reflection on the instrument surface; (c) instrument in the shadow; (f-g) specular reflection; (h-i) motion blur; (j) blood; (k) smoke; (l) bone; (m) titanium mesh; (n) gauze.

haven't been labeled in our dataset and the anatomical information has not been included in the proposed study on skill assessment, the analysis will be based on metrics of global rating.

2.1.3 Challenging Conditions

The main difficulties of this dataset include background reflections on instrument surfaces, specular reflections, blur from motion, blood and smoke. Figure 2.2 shows some examples of challenging frames. In both cadaveric and live surgery videos, reflections make the instruments have similar appearances to the background. In the live surgery videos, segmentation becomes even harder with instruments colored by blood. The endoscope and instruments are

usually handled at relatively low speeds during surgery, but senior surgeons who are more familiar with the anatomies may move tools faster, especially when switching surgery sites. The fast movements may lead to motion blur in video frames. Smoke generated when electrocauteries are used can also make the instruments hard to identify. Further, background objects like bone and gauze can be easily misclassified as instruments due to the colors they have.

Chapter 3

IMPROVING SEGMENTATION ROBUSTNESS WITH TEMPORAL INFORMATION

Currently, using surgery videos provided by endoscopic imaging equipment to guide medical treatment has become a common paradigm for many surgical interventions. As video-assisted surgery becomes more integrated, there is an increasing interest in surgical video analysis technologies. For the instrument segmentation task, videos contain instrument motion information that may help improve segmentation performance. As shown in Figure 2.2, it can be hard even for human eyes to identify instrument regions in surgical images with challenging conditions. When labeling some frames in the UW-Sinus-Surgery-C/L dataset, we had to refer to adjacent frames to decide the instrument regions. Considering this situation, we propose a model that can leverage temporal information from adjacent video frames for instrument segmentation.

3.1 Technologies for Analyzing Temporal Information

Recurrent Neural Networks (RNNs) are a typical type of networks for sequential data analysis and have been successfully applied in natural language processing [19, 79]. RNNs model temporal behaviors by recursively forwarding the previous states and using them together with current inputs to decide the outputs [124]. Long Short Term Memory (LSTM) and Gate Recurrent Unit (GRU) are two popular RNN architectures [124, 24]. RNNs have been widely adopted in many video analysis tasks such as video captioning, summarization, deblurring, achieving promising results [129, 128, 52, 89]. Additionally, 3D CNNs have been applied for video classification tasks, such as gesture recognition and skill assessment [127, 35, 36, 12]. In contrast, both RNNs and 3D CNNs have not been widely studied for

image segmentation, although intuitively, segmentation tasks on video frames could benefit from using temporal information. The complex gate mechanisms and the difficulty of training limit the use of RNNs in image segmentation [78, 123]. Also, RNNs usually require relatively long sequences for training, and this leads to a high computation burden, especially for computer vision problems. For 3D CNNs, although achieving promising performance with their rich spatiotemporal feature representation, they usually have more parameters and higher computation costs than 2D CNNs. As an alternative solution, researchers proposed to use spatial-temporal attention mechanisms to aggregate features from a few neighboring frames [49]. Another promising approach is to propagate the segmentation of the previous frame to assist segmentation on the current frame [118, 87].

Although deep learning has been successfully applied in surgical instrument segmentation as discussed in Section 1.1, most existing models perform segmentation on a single frame without leveraging the temporal information. Jin *et al.* proposed to propagate previous segmentation results based on motion flow to achieve more robust segmentation performance on the current frame [59]. Recently, spatial attention mechanisms that enhance feature representation by exploring the dependencies between pixels or feature channels have been explored for robust instrument segmentation [5, 90]. Although initial works have shown promising results, studies that leverage temporal and spatial information for instrument segmentation are still limited.

3.2 Multi-frame Feature Aggregation for Real-time Instrument Segmentation

In this work, we develop a Multi-frame Feature Aggregation (MFFA) module that aggregates features temporally and spatially for improving segmentation robustness. Section 3.2.1 provides an overview of the proposed approach. The MFFA module can be flexibly combined with general segmentation models that have an encoder-decoder architecture, allowing passing information from previous frames. The architecture of the MFFA module is described in Section 3.2.3.

In many public datasets for surgical instrument segmentation [105, 76], frames are ex-

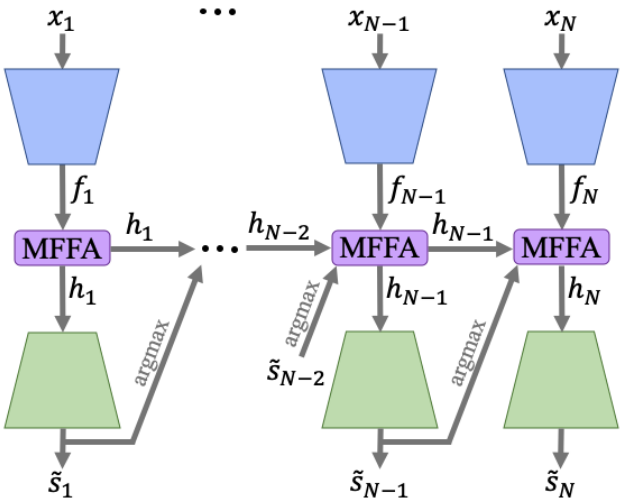

Figure 3.1: Overview of the proposed model. The MFFA module is embedded between the encoder (blue) and the decoder (green). The encoder-MFFA-decoder model works in a recurrent mode and this figure illustrates the unrolled structure over time steps.

tracted from videos with a certain sub-sampling rate for labeling, so a frame sequence from a surgical video usually has only one frame labeled. The sparsity of labeling may reduce the model convergence speed, especially in the early phase of training. To compensate for the lack of densely labeled real frame sequences, we initially investigate using synthetic sequences with every frame labeled for training, as described in Section 3.2.4. Each synthetic sequence consists of a real labeled frame and several frames synthesized based on this real frame.

3.2.1 Overview

Given an input sequence with N video frames $X = \{x_1, x_2, ..., x_N\}$, where x_i is the ith frame in the sequence. y_i is the corresponding ground truth segmentation map of x_i when x_i is labeled. Our goal is to train a model that can estimate the corresponding segmentation maps $Y = \{\tilde{y}_1, \tilde{y}_2, ..., \tilde{y}_N\}$, where $\tilde{y}_i$ indicates the estimated "instrument" and "background" regions of x_i.

The proposed framework is shown in Figure 3.1. Instead of segmenting each frame separately, we propose MFFA to aggregate features from the previous frame and pass the aggre-

gated features to the next frame. MFFA is designed to combine with general segmentation models that have an encoder-decoder architecture by inserting MFFA between the encoder and the decoder. The feature maps f_i from the encoder, the previous output of MFFA h_{i-1}, and the previous predicted segmentation mask $\tilde{y}_{i-1}$ are input to MFFA to generate aggregated feature maps h_i. h_i are then passed to the next frame and input to the decoder for segmenting the current frame. Finally, the encoder generates a softmax output $\tilde{s}_i$. The segmentation mask $\tilde{y}_i$ is then calculated with argmax from $\tilde{s}_i$.

Since MFFA is designed for general segmentation models, we demonstrate the proposed method based on a popular segmentation model DeepLabV3+ [18, 105, 97] with two representative backbone feature extractors, ResNet50 [46] and MobileNet [48]. These two backbones have been combined with different models and have achieved state-of-the-art performance in many recent works on instrument segmentation [105, 90, 97, 3, 108].

Moreover, because the feature maps are propagated and aggregated through frame sequences that usually consist of frames with a similar appearance, the proposed method could be considered as iterative feature aggregation. Therefore, we propose to use lightweight encoders to extract f_i. More specifically, we use block 1 to block 3 of ResNet50 as its trimmed version, named ResNet50-b3 [46]. We use the 1st to the 8th pointwise convolution of MobileNet as its trimmed version, named MobileNet-p8 [48].

3.2.2 Objective Functions

To train the proposed model, we extract real frame sequences from surgical videos. However, as discussed at the beginning of Section 3.2, sparsely labeled real frame sequences might reduce model training efficiency. Therefore, we propose a method that can generate synthetic frame sequences with every frame labeled (see Section 3.2.4) to explore if the model can benefit from training with densely labeled frame sequences. Specifically, we compare the models obtained in two settings: i) Train the model with real frame sequences; ii) Train the model with synthetic frame sequences in the first half of the training phase, and further train the model with only real frame sequences in the second half of the training phase.

For training, we calculate the cross-entropy loss to evaluate the segmentation performance for every $\tilde{s}_i$ that has ground truth y_i. After getting the one-hot encoding for y_i as s_i, the cross-entropy loss can be calculated as

$$\mathcal{L}_{CE}(s_i, \tilde{s}_i) = -\frac{1}{M} \sum_k (s_i)_k \log(\tilde{s}_i)_k \tag{3.1}$$

where M is the number of elements in the segmentation map, $(s_i)_k$ is the k-th element of s_i and $(\tilde{s}_i)_k$ is the k-th element of $\tilde{s}_i$.

When the network is trained with synthetic frame sequences, we input the sequence both forward and backward to the network and calculate the average cross-entropy loss of all frames. The first frame in a sequence does not have features passed from a previous frame, so only spatial feature aggregation can be performed on it (see Section 3.2.3). Therefore, the segmentation results and the losses will be different when passing a sequence forward and backward. To indicate whether a segmentation map is predicted with or without the information from a previous frame, we change the symbol of the predicted softmax output from $\tilde{s}_i$ to $\tilde{s}_i^j$. For a sequence of N frames, a $j \in [1, N]$ means the aggregated features of the j-th frame are used; otherwise, the segmentation is obtained without temporal information. Then the forward and backward average cross-entropy losses are given by

$$\mathcal{L}_{fw} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{CE}(s_i, \tilde{s}_i^{i-1}) \tag{3.2}$$

$$\mathcal{L}_{bw} = \frac{1}{N} \sum_{i=N}^{1} \mathcal{L}_{CE}(s_i, \tilde{s}_i^{i+1}) \tag{3.3}$$

where N is the number of frames in a sequence. When the network is trained with real frame sequences that have only the last frame labeled, we pass the sequence forward to the network and evaluate the segmentation accuracy on the last image. We use $\mathcal{L}_{last}$ to represent this loss and it is given by

$$\mathcal{L}_{last} = \mathcal{L}_{CE}(s_N, \tilde{s}_N^{N-1}) \tag{3.4}$$

Furthermore, when segmenting the first frame in a sequence during the training procedure, the aggregated features from the previous frame are not available. While during testing,

we pass the features through the whole video so most frames can be segmented based on sufficient temporal priors, as described in Section 3.3.1. Since training with long frame sequences may be impractical due to limited computation ability, it is important to ensure the network can extract good enough features from the first frame, so that the network is trained under closer situations to testing. Therefore, for each synthetic or real sequence, we input its real, labeled frame into the model without previous features, and calculate the cross-entropy loss $\mathcal{L}_{1st}$. Assume x_k is the real labeled frame in a sequence, then we have

$$\mathcal{L}_{1st} = \mathcal{L}_{CE}(s_k, \tilde{s}_k^0) \tag{3.5}$$

Finally, when the synthetic sequences are used for training, the overall objective function is given by

$$\mathcal{L} = \lambda_1 \mathcal{L}_{fw} + \lambda_2 \mathcal{L}_{bw} + \lambda_3 \mathcal{L}_{1st} \tag{3.6}$$

When the real sequences are used for training, the overall objective function is given by

$$\mathcal{L} = \lambda_4 \mathcal{L}_{last} + \lambda_5 \mathcal{L}_{1st} \tag{3.7}$$

where λ_i are hyper-parameters that balance the impact of the losses. In this work, we choose $\lambda_1 = \lambda_2 = \lambda_3 = \frac{1}{3}$ and $\lambda_4 = \lambda_5 = \frac{1}{2}$.

3.2.3 Multi-frame Feature Aggregation (MFFA) Module

The MFFA module consists of a Temporal Aggregation Block (TAB) and a Spatial Aggregation Block (SAB) in series as shown in Figure 3.2. MFFA takes $\tilde{y}_{i-1}$, h_{i-1} and f_i as inputs and outputs h_i. In endoscopic sinus surgery, distinguishing instruments from the background can be very challenging as shown Section 2.1.3. Because instrument locations are usually close in neighboring video frames, in TAB, we emphasize the instrument features of f_i by aggregating with the instrument features of h_{i-1}. The instrument features are extracted by element-wise multiplication between the previous output y_{i-1} and h_{i-1}. The extracted previous instrument features and current features are then concatenated and passed to two parallel 1x1 convolutions. One of the convolutions performs feature aggregation. The other

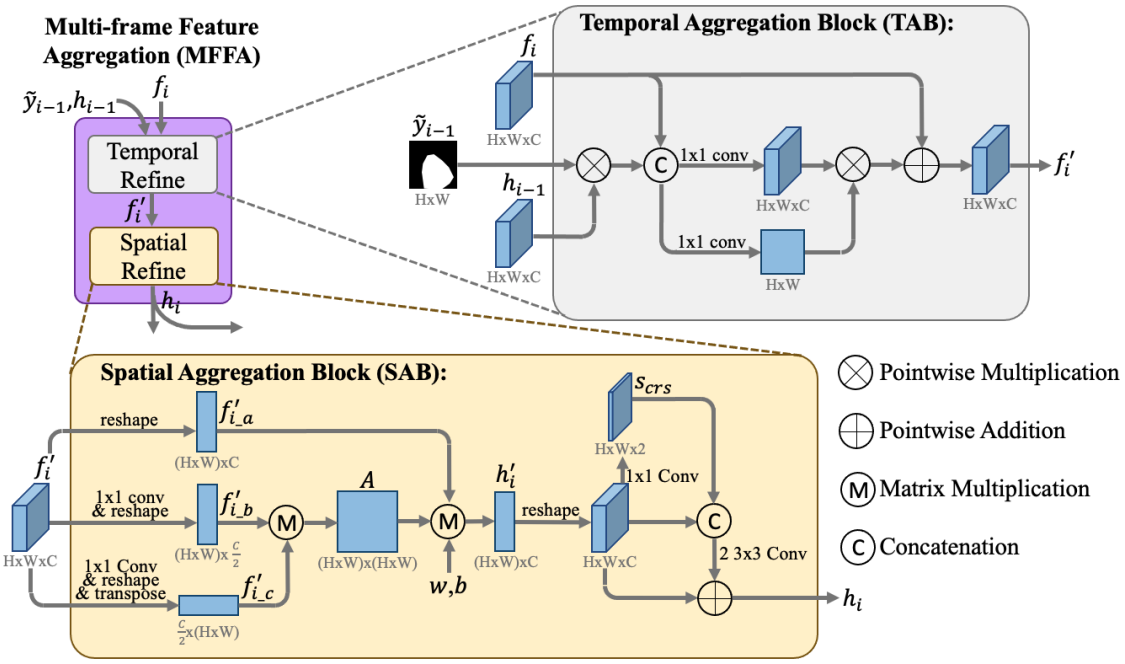

Figure 3.2: Schematic of the proposed MFFA module. The MFFA module (purple) consists of a Temporal Aggregation Block (gray) and a Spatial Aggregation Block (yellow).

convolution is followed by a sigmoid function to estimate the similarity between the previous and current feature maps and give a 2D weight map that serves as a gate to regulate the aggregated features. The aggregated features are then multiplied with the corresponding weights and finally added by f_i to generate f_i' as the output of TAB. When applying MFFA to the first frame in a sequence, the previous output and features are not available, so we skip TAB and directly input f_i to SAB.

After temporal feature aggregation in TAB, the generated features f_i' is further aggregated using SAB based on spatial relationships. SAB is partially inspired by the spatial self-attention module [34] and Graph Attention Network (GAT) [82]. The input features of SAB $f_i' \in \mathbb{R}^{H \times W \times C}$ can be considered as a graph with $H \times W$ vertices. First, we use part of the spatial self-attention module to calculate a spatial attention matrix $A \in \mathbb{R}^{(H \times W) \times (H \times W)}$ [34]. Specifically, f_i' is passed to two parallel 1×1 convolutions for generating two new feature

maps $\{f'_{i\_b}, f'_{i\_c}\} \in \mathbb{R}^{H\times W\times \frac{C}{2}}$, which are then reshaped to $\mathbb{R}^{(H\times W)\times \frac{C}{2}}$. Next we transpose f'_{i_c} to $\mathbb{R}^{\frac{C}{2}\times(H\times W)}$ and perform matrix multiplication between f'_{i_b} and f'_{i_c} to get the spatial attention matrix A. Each element of A represents the similarity between the corresponding vertices in graph f'_i. Then we reshape f'_i to $f'_{i\_a} \in \mathbb{R}^{(H\times W)\times C}$ and aggregate the vertices features using A through matrix multiplication. After that we use trainable parameters $w \in \mathbb{R}^{C\times C}$ and bias $b \in \mathbb{R}^C$ to get a refined features h'_i. The expression of this process is given by

$$h'_i = Af'_{i\_a}w + b \tag{3.8}$$

Next, we reshape h'_i from $\mathbb{R}^{(H\times W)\times C}$ to $\mathbb{R}^{H\times W\times C}$ and further refine the features using a ResNet module-like block as proposed in [126]. Specifically, h'_i goes through a 1×1 convolution and the softmax function to generate a coarse segmentation results $\tilde{s}_{crs}$. Finally, the output of SAB is obtained by

$$h_i = h'_i + \Phi(h'_i, \tilde{s}_{crs}) \tag{3.9}$$

where $\Phi(a, b)$ performs a and b concatenation followed by two 3×3 convolution filters in series.

3.2.4 Synthetic Frame Sequence

We propose a method that can generate a synthetic frame sequence from a real labeled frame. The proposed sequence synthesis method is developed based on the data augmentation method introduced in [30]. In [30], the dataset was augmented by cropping and randomly jittering the target objects in images. The holes left by the moved target objects were then filled with an off-the-shelf inpainting method [112].

To add temporal information to the method of [30], we synthesize frame sequences for training. Given a real frame x and its ground truth segmentation map y, our goal is to generate a frame sequence with N labeled frames $Z = \{z_1, z_2, ..., z_N\}$. We first put x to the center of the target sequence, i.e. let $z_C = x$ where $C = \lfloor \frac{N+1}{2} \rfloor$. For the instrument region in the ith frame, we define its translations on the x and y-axis and the rotation with respect

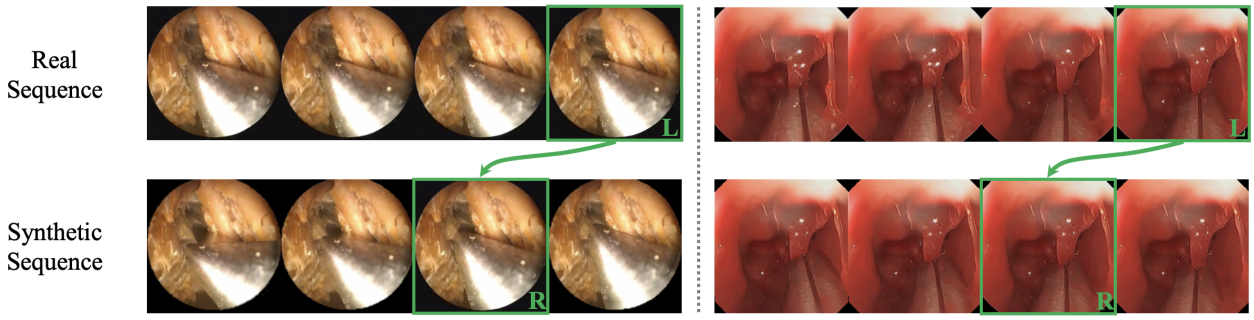

Figure 3.3: Examples of real and synthetic sequences. The top row shows two real sequences, each has only one labeled frame, denoted by 'L' in its bottom right corner. The second row shows the synthetic sequences generated from the corresponding labeled real frames in the first row. The real frame that each synthetic sequence is generated from is denoted by 'R' in its bottom right corner, while other frames in the synthetic sequence are augmented from the real frame. Refer to Section 2 for more information of the datasets.

to the instrument region in x as dx_i, dy_i and $d\theta_i$, which are called moving parameters. Then we generate the moving parameters for the first frame, i.e., dx_0, dy_0, $d\theta_0$, from a uniform distribution (see Section 3.3.1 for more details). Next, we use linear interpolation to decide the moving parameters of the other frames in the target sequence. Finally, the instruments are cropped, rotated, and moved to the corresponding locations using the aforementioned augmentation method to generate all synthetic frames that form the target sequence. We also apply the same process with the same moving parameters on y to get the corresponding labels. Figure 3.3 shows examples of synthetic frame sequences. Figure 3.3 also shows the corresponding real frame sequences that include the real frames used to generate the synthetic frame sequence.

3.3 Experiments and Results

We combined the proposed MFFA module with DeepLabV3+ (DL3+) [18] (abbreviated as MFFA-DL3+) and compared with the original version of DeepLabV3+ [18] as baseline. MFFA-DL3+ were also compared with advanced models that perform segmentation on single

frames, including TernausNet [53], LWANet [90], and MAFA [97].

3.3.1 Implementation Details

The segmentation models were implemented on a 4.20GHz Intel i7-7700K CPU and a Nvidia Titan Xp GPU.

Datasets

We conducted experiments on the UW-Sinus-Surgery-C/L dataset through 3-fold cross-validation (see details in Section 2.1.2). We also evaluated the models on a public dataset ROBUST-MIS [104, 85]. Because the labels of the test set are not available, we used the training set of the proctocolectomy dataset in ROBUST-MIS for the binary segmentation task. The proctocolectomy training set consists of 2943, 960×540 labeled frames sampled from 8, 3-5 hour laparoscopic videos corresponding to 8 procedures. Similarly, we performed 3-fold cross-validation for evaluation. Based on the number of frames from each patient, we split the given training data into the following 3 folds: 6 videos (Procedure ID: 1, 2, 3, 4, 5, and 8) with 851 labeled frames, the video of Procedure 9 with 958 labeled frames, and the video of Procedure 10 with 1134 labeled frames. In each of the three validations, we used two folds for training and the remaining one fold for testing.

Instrument Segmentation Models

The segmentation models were implemented with or without the proposed MFFA module. The backbones of all models were pre-trained on ImageNet [28]. MFFA was implemented based on Figure 3.2 by setting $C = 128$ and adding a ReLU activation function at the output of TAB and after all convolutions in SAB. For UW-Sinus-Surgery-C/L, we used the Adam optimizer [63] to train each model with 40 epochs and 16 batch size. The learning rate was initialized as 0.0005 and exponentially decayed every 5 epochs from the 20th epoch with a decay rate of 0.5. To evaluate the models on ROBUST-MIS-proctocolectomy, we chose a

batch size of 8 while keeping the epoch and learning rate settings the same.

When we trained the models without MFFA, we augmented the images by i) changing hue, brightness, saturation and contrast; ii) flipping, rotation, scaling, and cropping. When using UW-Sinus-Surgery-C/L, we cropped the frames to a resolution of 192×192 to accelerate training, while the full images with a resolution of 240×240 were used for testing. For ROBUST-MIS-proctocolectomy, the raw images were resized to 640×360 and kept in the same size after augmentation; during testing, the raw images were resized to 640×360 for inference, and then the output segmentation maps were resized back to 960×540 to evaluate the performance. When we trained the models with MFFA, we implemented similar data augmentation on the frame sequences. We jittered the appearance of every frame in the sequences independently, while all frames in the same sequence shared the same parameters for flipping, rotation, scaling, and cropping.

Additionally, on UW-Sinus-Surgery-C/L, we compared the models (with MFFA) that were either: i) trained with real frame sequences or ii) trained with synthetic frame sequences in the first 20 epochs and trained with real frame sequences in the last 20 epochs. Details of these two training settings can be found in Section 3.2.2. The second setting is represented with 'Synth.' in Table 3.1 and Table 3.3. For ROBUST-MIS-proctocolectomy, we only used the real frame sequences for training.

Synthetic and Real Frame Sequence

Considering the available computation ability, we trained models with sequences of 4 frames. To generate synthetic sequences, we need to specify the translations and rotation angles of the instruments in the first frame with respect to the instrument in the labeled frame. The translation values were randomly selected from a uniform distribution over 15 to 40 pixels on both positive/negative x-axis and positive/negative y-axis directions. The rotation angles were randomly selected from a uniform distribution over -40 to 40 degrees. The parameters of these uniform distributions were selected empirically to generate synthetic sequences that resembled motion in the videos. In addition, real frame sequences were extracted around

labeled frames in the videos and each frame in a sequence was sampled out of 3 consecutive frames.

3.3.2 Experiments on UW-Sinus-Surgery C/L

Evaluation

The segmentation performance was evaluated based on Dice similarity coefficient (DSC) and Intersection over Union (IoU) [109], which are defined as

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} \tag{3.10}$$

$$IoU = \frac{|X \cap Y|}{|X \cup Y|} \tag{3.11}$$

where X and Y are the predicted and ground truth segmentation maps, respectively. Further, we estimated the computation cost of each model using its inference time, which is the time for the model to predict a segmentation map.

The models were evaluated with 3-fold cross-validation as described in Section 2 and a video can only be used either for training or testing in each of the three validations. The trained models were tested by propagating the aggregated feature maps and performing segmentation every third frame throughout the test videos, and the segmentation performance was evaluated on the labeled frames.

Results

The segmentation results on UW-Sinus-Surgery C/L is shown in Table 3.1. There are three groups of methods: i) Group 1 consists of three advanced segmentation models on single frames; ii) Group 2 consists of the original version of DL3+(MobileNet), DL3+(MobileNet-p8), and MFFA-DL3+(MobileNet-p8) trained with/without synthetic sequences; iii) Group 3 consists of the original version of DL3+(ResNet50), DL3+(ResNet-b3), and MFFA-DL3+(ResNet-b3) trained with/without synthetic sequences. To determine if there are significant

Table 3.1: Segmentation performance on UW-Sinus-Surgery-C/L

No.	Model(Backbone)	MFFA	Synth.	Performance (mDSC(%) / mIoU(%))		Time
				Sinus-Surgery-C	Sinus-Surgery-L	(ms)
1	TernausNet-16 [53] (VGG16)	n/a	n/a	85.4(2.2)/80.1(2.7)	79.5(5.9)/73.4(6.8)	12.9
	LWANet [90] (MobileNet)			81.1(3.8)/74.9(4.5)	72.6(7.7)/65.2(8.8)	5.3
	MAFA-DL3+ [97] (ResNet50)			91.2(1.0)/86.8(1.2)	87.7(3.8)/82.1(4.6)	19.3
2	DL3+ [18] (MobileNet)	×	×	81.4(4.2)/75.5(4.8)	76.6(6.8)/69.5(8.0)	3.1
	DL3+ [18] (MobileNet-p8)	×	×	80.4(3.5)/74.0(4.4)	75.7(7.0)/68.0(8.3)	2.1
		✓	×	84.8(2.9)/79.4(3.3)	81.3(6.0)/74.7(7.1)	2.9
		✓	✓	**86.4(2.2)/81.1(2.7)**	**83.2(4.1)/77.0(5.0)**	2.9
3	DL3+ [18] (ResNet50)	×	×	86.0(2.0)/81.0(2.5)	81.8(5.4)/75.2(6.4)	8.2
	DL3+ [18] (ResNet50-b3)	×	×	83.9(5.0)/78.8(5.6)	80.1(5.4)/73.4(6.4)	5.0
		✓	×	86.1(2.5)/81.0(3.0)	83.2(4.9)/77.0(5.7)	5.4
		✓	✓	**88.9(1.4)/84.0(1.8)**	**85.8(3.3)/80.0(4.3)**	5.5

* i) The mDice and mIoU are shown in the form of 'mean(standard deviation)'; ii) MobileNet-p8 is the trimmed MobileNet; iii) ResNet50-b3 is the trimmed ResNet50; iv) Both synthetic and real data are used for training if 'Synth.' is checked, otherwise only real data are used for training; v) The bold font indicates the best performance in the column of each group.

differences between the eight models in Group 2 and 3, we performed Friedman test [33] followed by a Conover post hoc test [25] on the 6 data folds of UW-Sinus-Surgery C/L. Friedman test shows that there is a statistically significant difference between the performance of these eight models with a p-value less than 0.05 and the pairwise comparisons calculated using Conover's test are shown in Table 3.2. In each row of Table 3.2, a p-value less than 0.5 indicates that there is a statistically significant difference between the corresponding paired comparison methods.

When both synthetic and real frame sequences were used for training, MFFA-DL3+ with reduced backbone achieved superior performance of 3.1%∼9.5% better mDice and 3.3%∼10.7% better mIoU with less inference time compared with the baseline DL3+. The

Table 3.2: Conover post hoc test results of baselines (DeepLabV3+ [18]) and proposed models on UW-Sinus-Surgery-C/L Dataset

Compared Model (Backbone,**R**eal/**S**ynthetic Data)		p-value
DL3+(MobileNet,R)	DL3+(MobileNet-p8,R)	0.83
	MFFA-DL3+(MobileNet-p8,R)	0.08
	MFFA-DL3+(MobileNet-p8,S)	0.02*
DL3+(ResNet50,R)	DL3+(ResNet50-b3,R)	0.38
	MFFA-DL3+(ResNet50-b3,R)	0.74
	MFFA-DL3+(ResNet50-b3,S)	0.07
	MFFA-DL3+(MobileNet-p8,S)	0.83
MFFA-DL3+ (MobileNet-p8,R)	MFFA-DL3+ (MobileNet-p8,S)	0.51
MFFA-DL3+ (ResNet50-b3,R)	MFFA-DL3+ (ResNet50-b3,S)	0.13

* i) In the brackets, 'R' represents only real data are used for training, while 'S' represents both synthetic and real data are used for training; ii) In the last column, p-values less than 0.05 are marked with '*'.

third row of Table 3.2 further shows that there is a statistically significant difference between MFFA-DL3+(MobileNet-p8) trained with synthetic frame sequences and DL3+(MobileNet). Also, MFFA-DL3+(MobileNet-p8) achieved comparable results to DL3+(ResNet50) with a p-value of 0.83 while used only about 35% of the average inference time. Moreover, compared with the three advanced segmentation models shown in Group 1, both MFFA-DL3+(MobileNet-p8) and MFFA-DL3+(ResNet-b3) achieved superior or comparable performance with much lower computation costs.

To evaluate the effectiveness of i) Temporal Aggregation Block (TAB), ii) Spatial Aggregation Block (SAB), and iii) $\mathcal{L}_{1st}$, we conducted ablation studies on UW-Sinus-Surgery C/L with DeepLabV3+ [18], as shown in Table 3.3. The original version of DL3+(MobileNet) was used in experiment 1, while DL3+ with a trimmed backbone MobileNet-p8 was used in ex-

Table 3.3: Ablation studies of MFFA with DeepLabV3+ on UW-Sinus-Surgery-C/L Dataset

No.	Method					Performance (mDSC(%) / mIoU(%))	
	Backbone	TAB	SAB	Synth.	$\mathcal{L}_{1st}$	Sinus-Surgery-C	Sinus-Surgery-L
1	MobileNet	×	×	×	×	81.4(4.2)/75.5(4.8)	76.6(6.8)/69.5(8.0)
2	MobileNet-p8	×	×	×	×	80.4(3.5)/74.0(4.4)	75.7(7.0)/68.0(8.3)
3		×	✓	×	×	82.4(1.7)/76.6(2.2)	79.1(5.4)/71.7(6.6)
4		✓	×	×	×	83.0(3.3)/76.6(4.0)	77.4(7.0)/69.9(8.4)
5		✓	✓	×	×	83.8(1.6)/78.2(2.0)	80.1(5.5)/73.2(6.6)
6		✓	✓	×	✓	84.8(2.9)/79.4(3.3)	81.3(6.0)/74.7(7.1)
7		✓	✓	✓	×	84.3(1.7)/78.9(2.3)	79.8(5.3)/73.3(6.2)
8		✓	✓	✓	✓	**86.4(2.2)/81.1(2.7)**	**83.2(4.1)/77.0(5.0)**

* i) Both synthetic and real data are used for training if 'Synth.' is checked, otherwise only real data are used; iii) The bold font indicates the best performance in the column.

periment 2∼8. Because TAB is responsible for aggregating temporal information, the model was trained with frame sequences only when TAB was used (experiment 4-8). Compared with experiment 2, experiment 3 shows that using SAB without temporal feature aggregation improved the performance on both the cadaver and live datasets, while experiment 4 shows that TAB improved the segmentation performance by leveraging the temporal information. Further, experiment 5 shows that better performance was achieved by using TAB and SAB together. By comparing experiments 5 and 6, and comparing experiments 7 and 8, we found $\mathcal{L}_{1st}$ is more effective when training with synthetic frame sequences.

Further, we assessed the sensitivity of segmentation performance to the frame sequence length on UW-Sinus-Surgery-C with MFFA-DL3+(MobileNet-p8). Figure 3.4 shows the test mDSC with frame sequence length ranging from 2 to 8. We found that training the model with synthetic sequences achieved better performance than when only real sequences are used. However, limited by the computation resource, the model cannot be tested on longer sequences, which are needed to draw a conclusion on the relationships between sequence length and segmentation performance.

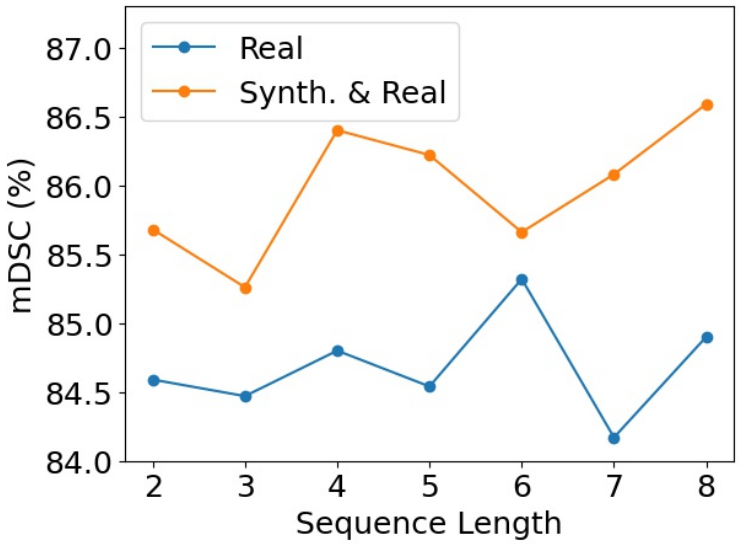

Figure 3.4: mDSC achieved by MFFA-DL3+(MobileNet-p8) on UW-Sinus-Surgery C/L with sequences of different lengths. The blue line ('Real') shows the performance of models trained only with real sequences. The yellow line ('Synth. & Real') shows the performance of models trained with both synthetic and real sequences. In each real sequence, ground truth is only known for one frame of the sequence, while every frame in the synthetic sequence is labels.

3.3.3 Experiments on ROBUST-MIS-Proctocolectomy

Table 3.4 shows the segmentation results on ROBUST-MIS-Proctocolectomy. We compared the performance of DL3+(MobileNet), DL3+(MobileNet-p8), and MFFA-DL3+(MobileNet-p8) with using only real frame sequences for training. Compared with the original version of DL3+(MobileNet), MFFA-DL3+(MobileNet-p8) achieved superior performance of 2.5%∼3.0% better mDice and 3.3%∼3.8% better mIoU with less inference time.

3.4 Discussions and Conclusions

In this work, we developed and validated a MFFA module that performs feature aggregation based on temporal and spatial relationships between frame pixels to improve instrument segmentation. By using the MFFA module, we can reduce the deep encoder to its trimmed version and decrease the computation costs. Another advantage of the proposed MFFA

Table 3.4: Segmentation performance on ROBUST-MIS-Proctocolectomy

Model(Backbone)	mDSC (%)	mIoU (%)	Time (ms)
TernausNet-16 [53] (MobileNet)	79.6(2.9)	71.6(3.0)	21.6
LWANet [90] (MobileNet)	76.2(3.2)	67.6(3.3)	14.2
MAFA-DL3+ [97] (MobileNet)	81.7(2.8)	74.1(3.0)	27.8
DL3+ [18] (MobileNet)	78.1(3.1)	69.9(3.2)	9.0
DL3+ [18] (MobileNet-p8)	78.4(3.8)	70.4(4.0)	6.4
MFFA-DL3+ [18] (MobileNet-p8)	**81.0(3.1)**	**73.5(3.2)**	8.5

* i) The mDSC and mIoU are shown in the form of 'mean(standard deviation)'; ii) The bold font indicates the best performance in the column of the second group (last three rows).

module is that it can be easily combined with any segmentation model that has an encoder-decoder architecture. Also, we proposed a simple but effective strategy that can generate a synthetic frame sequence from a single labeled frame to assist network training and compensate for a lack of labeled real frame sequences. The experiment results demonstrated that by combining the proposed feature aggregation module MFFA with an existing segmentation model, we achieved promising segmentation performance with low computation costs.

Figure 3.5 shows some examples of the segmentation results. As shown in Figure 3.5(a-d,f), the proposed MFFA module could help improve instrument segmentation by reducing false negatives under many challenging conditions including reflection and blood compared to the baseline experiments. Moreover, we found that the MFFA module could help reduce the false positives on objects that are not parts of the human body such as gauze as shown in Figure 3.5(e). In this work, we focused on semantic segmentation in which pixels are classified into different classes without separating different objects. When analyzing surgical videos that have several instruments operating at the same time, instance segmentation that treats different objects of the same class separately is more appealing as it provides more information regarding the surgical workflow. We will explore leveraging temporal information

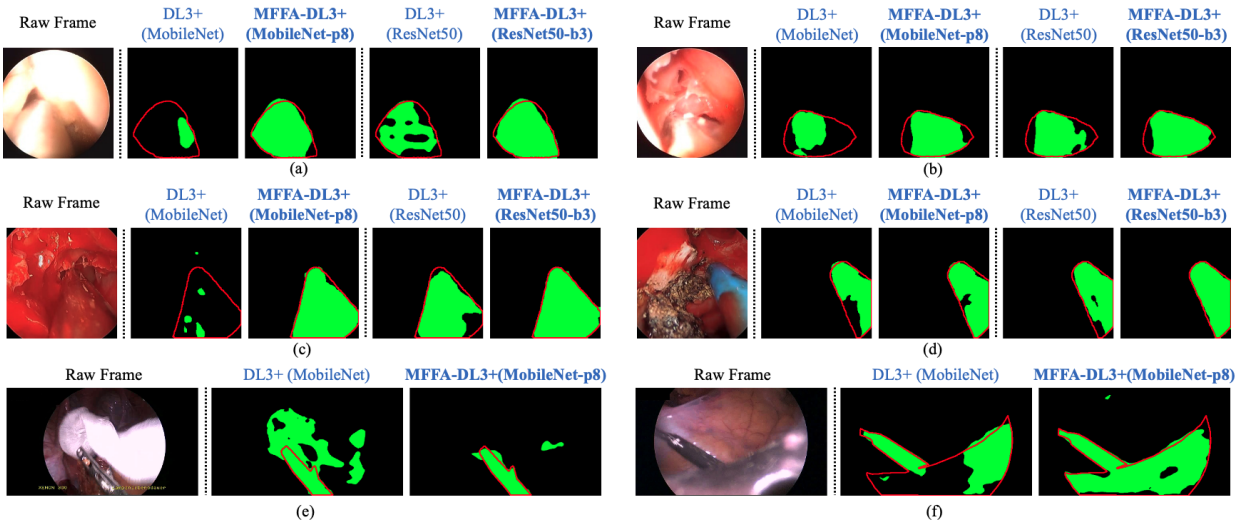

Figure 3.5: Examples of segmentation results. The result samples of UW-Sinus-Surgery-C, UW-Sinus-Surgery-L, ROBUST-MIS-Proctocolectomy are shown in (a,b), (c,d), and (e,f), respectively. In (a-d), the first frame is the input raw frame and the next four frames are the segmentation results of DeepLabV3+ (DL3+) [18] with MobileNet as the backbone feature extractor, DL3+ with MobileNet-p8 and MFFA, DL3+ with ResNet50, and DL3+ with ResNet50-b3 and MFFA, respectively. In (e-f), the first frame is the input raw frame and the next two frames are the segmentation results of DL3+ with MobileNet, and DL3+ with MobileNet-p8 and MFFA, respectively. MFFA-DL3+(MobileNet-p8) and MFFA-DL3+(ResNet50-b3) were trained with both synthetic and real frame sequences. The predicted instrument regions are drawn in green and the true instrument contours are labeled by red lines.

for instance segmentation as future work.

Table 3.1 shows that by using the synthetic frame sequences we achieved better segmentation performance than using only the real frame sequences, but Table 3.2 shows that this superiority is still not significant enough with p-values of 0.51 and 0.13. We think there is still some space left for improvement because the current synthesis method is relatively simple. We used a simplistic instrument trajectory model for two reasons: 1) the frame sequences used in this work have only 4 frames with small instrument movements, so this method provides visually realistic approximations to the real sequences. 2) The statistical or other knowledge of instrument moving behaviors are not available in the datasets studied in

this work. The true instrument movement pattern could be considered in the future to generate more realistic sequences. One potential option is performing interpolation or modeling using neighboring labeled frames to generate better synthetic trajectories.

Chapter 4

IMPROVING GENERALIZABILITY OF INSTRUMENT SEGMENTATION MODELS WITH TRANSFER LEARNING

As deep learning has achieved more state-of-the-art performance in many engineering problems, the demands for large datasets are increasing. While collecting raw unlabeled data becomes easier, data labeling is still expensive and time-consuming. Labeling medical data often requires some medical knowledge so it is hard to collect a reasonable amount of data for training [22, 10]. On the other hand, in surgical images, the appearance of the instrument and background varies a lot due to different illumination conditions, various types of organs or tissues, and the presence of blood and smoke. Moreover, the usage frequencies of some instruments are related to surgeons' skill levels [58]. During endoscopic sinus surgery, surgeons hold the instruments with one hand and hold the endoscope with the other hand, so the instrument size in surgical images depends on the surgeons' habit of holding the instruments and endoscope. Therefore, a segmentation model trained on a dataset collected from a specific hospital usually does not generalize well to datasets collected from different locations even the surgical type remains the same, let alone transfer to a different type of intervention. If labeling a large amount of training data is needed whenever dealing with a new dataset, the application of deep learning in instrument segmentation will be greatly impeded. Transfer learning aims to improve the task performance on target domains by transferring knowledge learned from different but related domains (called source domains) [134, 22]. Success in transfer learning could reduce the need for data labeling. This property makes transfer learning appealing for clinical applications.

4.1 State-of-the-art Domain Adaptation Technologies

In transfer learning, both the tasks and data appearance and distribution of source and target domains can be different. In medical image analysis, fine-tuning models pretrained on large-scale datasets (*e.g.*, ImageNet [28]) with target data has become a common approach to accelerate model training and remedy the influence of limited data [134, 45]. The reader is referred to [134] for a review on transfer learning with pretrained models. Our work focuses on a particular case of transfer learning called domain adaptation, where the task remains the same but there is a gap between the source and target domain [64, 94]. Many efforts on domain adaptation for semantic segmentation have been made to align different domains in the feature space via adversarial learning [20, 114, 132]. This method usually involves discriminators that can distinguish the differences between features of the source and target data. Training discriminators with the task model shared across domains encourages the model to find the common feature space and allows better task performance on target domain.

Aligning different domains in pixel-level using image translation methods, such as Generative Adversarial Networks (GANs) [42] and CycleGAN [133], is another popular strategy [21, 69, 76, 84]. GANs consist of two networks, a generator and a discriminator, which are trained to contest with each other. More specifically, the discriminator is trained to distinguish the translated data from the real data and the generator is trained to fool the discriminator. CycleGAN has two GANs to translate data bidirectionally between the source and target domain and uses a cycle consistency loss to ensure the translation is invertible. Through image translation, the data are brought to the same domain, and therefore, allowing training shareable task models with the labeled data of the source domain.

In semi-supervised learning and self-supervised learning, training networks on unlabeled data with high-confident pseudo labels in a supervised fashion has led to advanced performance [101, 68, 54, 125]. This strategy has also been applied to domain adaptation as a complementary to domain alignment. For image classification, a popular method is to al-

ternately 1) perform clustering to assign target data pseudo labels based on the source data in the corresponding clusters and 2) optimize networks with source data labels and target data pseudo labels [60, 43]. In [69], the researchers proposed to use pseudo labels of target data and ground truth labels of translated source data (i.e., data translated from the source domain to the target domain) together to train models.

4.2 LC-GAN: Exploration on Pixel-level Domain Alignment

We explore image-to-image translation that aims to align domains in the pixel level, so that the knowledge could be transferred between different domains. This work is under the unsupervised domain adaptation (UDA) setting for the instrument segmentation task, i.e., perform segmentation on the unlabeled data in the target domain by leveraging the model trained with the labeled data in the source domain. The two datasets used here are the cadaveric and live dataset of UW-Sinus-Surgery-C/L (see Section 2). Although we have manually labeled both datasets, we assume the labels of the cadaveric dataset are available while the labels of the live dataset are only used to evaluate the proposed method.

4.2.1 Overview

The schematic of the proposed image-to-image translation model live-cadaver GAN (LC-GAN) is shown in Figure 4.1 (a). LC-GAN is developed to learn the mapping between the cadaveric dataset and live dataset. We then perform instrument segmentation on the fake-cadaveric surgery images generated from the real-live surgery images, as shown in Figure 4.1 (b). The segmentation is implemented using deep convolutional neural network (CNN) models trained with the labeled cadaveric dataset. Finally, we evaluate the potential of the proposed method by comparing its segmentation performance with the traditional method that trains and tests the segmentation model in the same domain.

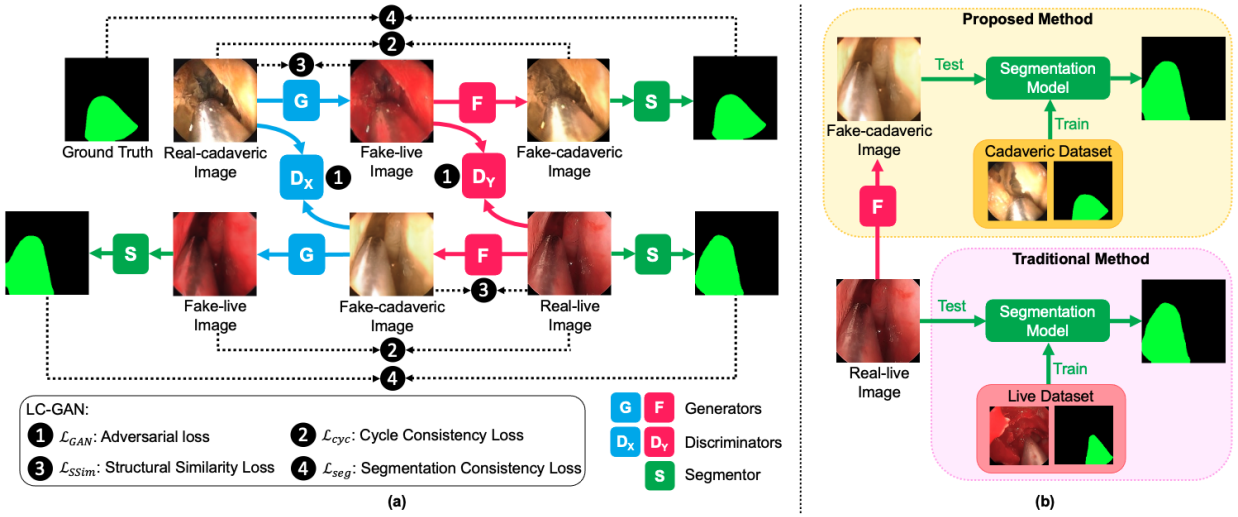

Figure 4.1: Overall framework. (a) Schematic of LC-GAN. The generator G performs cadaver-to-live translation, while generator F performs live-to-cadaver translation. The discriminators D_X and D_Y are used to distinguish the fake images from the real images in cadaveric and live domains, respectively. The segmentor S is a deep segmentation model pre-trained on the real cadaveric dataset. (b) Schematic of the proposed method and the traditional method for instrument segmentation on the live dataset. The instrument regions in the segmentation maps are shown in green. Note that there is only one instance of G, F, and S, respectively.

4.2.2 Live-cadaver GAN (LC-GAN) Architecture

Similar to CycleGAN [133], LC-GAN consists of two generators and two discriminators. Given two image domains X and Y, the two generators G and F are trained to learn the mappings between the image domains, i.e. $G : X \rightarrow Y$ and $F : Y \rightarrow X$. D_X and D_Y are two discriminators trained to distinguish between real and fake images. We define the cadaveric dataset as the X domain and the live dataset as the Y domain.

We construct two generators with different architectures as shown in Figure 4.2. Since deep segmentation models can be trained on the cadaveric dataset with labels, we leverage backbone feature extractors of the trained segmentation models to provide better feature maps for the generator G that maps cadaveric surgery images to live surgery images. We also adapt part of the DeepLabV3+ segmentation model [18] into G. Specifically, we apply

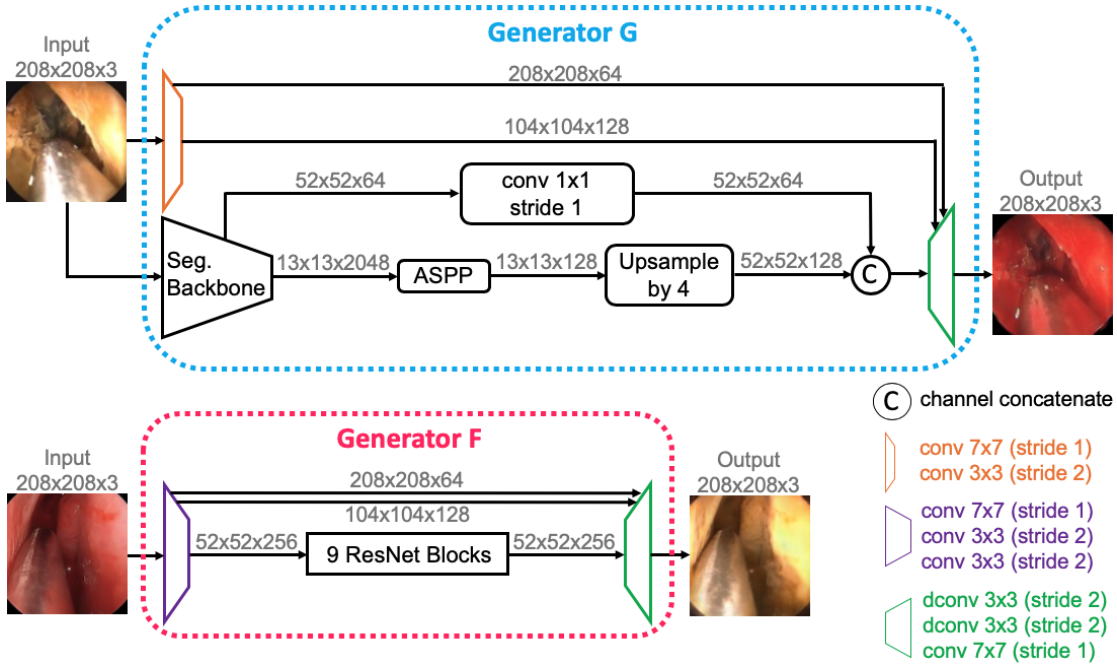

Figure 4.2: Generator architectures of LC-GAN (example for 208x208 input image). Each trapezoid represents a series of convolution or deconvolution operations. The sizes (width-height-channel) of the feature maps are shown on top of the corresponding arrows.

the Atrous Spatial Pyramid Pooling (ASPP) module [18] to extract multi-scale features from the output of pre-trained backbones, and the extracted features are then concatenated with low-level features of the backbones. The parameters of the pre-trained backbones are fixed during LC-GAN training. For another generator F, no trained feature extractor is available, so we train it from scratch. We choose a ResNet with two stride-2 convolutions, nine residual blocks, and two fractionally-strided convolutions with stride $\frac{1}{2}$ as the generator F [46, 133]. For the discriminator, we use the 70×70 PatchGAN [56, 133] to distinguish the real 70×70 image patches from fake patches.

4.2.3 Loss Functions for LC-GAN

CycleGAN was proposed with the adversarial loss $\mathcal{L}_{GAN}$ and the cycle consistency loss $\mathcal{L}_{cyc}$. The details of $\mathcal{L}_{GAN}$ and $\mathcal{L}_{cyc}$ are provided in [133]. Although CycleGAN shows compelling

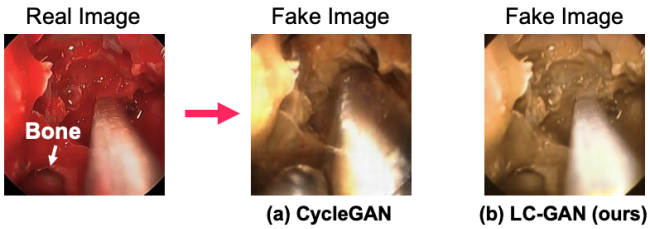

Figure 4.3: Translation from a real-live surgery image to a fake-cadaveric surgery image. (a) The result of CycleGAN is an example of semantic inconsistency. The bone region pointed by the white arrow is translated into an instrument and the instrument becomes much larger in the fake image than its true size. (b) The proposed LC-GAN generates a fake image with better semantic consistency.

results in many datasets, the semantic consistency might not be guaranteed for some complex scenes. Figure 4.3(a) shows an example of semantic inconsistency between the real-live surgery image and the fake-cadaveric surgery image generated using CycleGAN. In Figure 4.3(a), the bone is translated to an instrument and the true instrument becomes much larger in the fake image. To ensure the semantic consistency, we propose a structural similarity loss $\mathcal{L}_{SSim}$ and a segmentation consistency loss $\mathcal{L}_{seg}$ alongside with the cycle consistency loss as shown in Figure 4.1(a).

$\mathcal{L}_{SSim}$ estimates the structural similarity between the input and output of a generator. Because the live and cadaveric surgery images have very different colors, the color information should be excluded when calculating $\mathcal{L}_{SSim}$. Therefore, we convert the image from RGB to YUV color space and use the Y channel that consists of luminance information [32] to estimate the structural similarity. Besides color features, the live surgery images have different lighting conditions and the presence of fluids including blood compared to the cadaveric surgery images. Therefore, $\mathcal{L}_{SSim}$ should focus on overall structural similarity while allowing differences in image brightness, contrast and details. Similar to [120, 95], we compare the image structural information at different resolutions. Specifically, we iteratively downscale the image by a factor of 2 and get a total of n_s scaled images which are 1, $\frac{1}{2}$, ...,

$\frac{1}{2^{n_s}}$ of the original size, respectively. The expression of $\mathcal{L}_{SSim}$ is

$$\begin{aligned}\mathcal{L}_{SSim}(G,F) = [1 - \sum_{i=0}^{n_s} \gamma_i C(x_{Y,i}, G(x)_{Y,i})] \\ + [1 - \sum_{i=0}^{n_s} \gamma_i C(y_{Y,i}, F(y)_{Y,i})]\end{aligned} \tag{4.1}$$

where $x \in X$ is an image from the X domain and $y \in Y$ is an image from the Y domain, the subscripts Y, i of an image denote that we extract the Y channel of the image and scale it to $\frac{1}{2^i}$ of its original size, γ_i are the multi-scale weights and are normalized to $\sum_{i=0}^{N} \gamma_i = 1$. $C(a, b)$ is the zero-normalized cross-correlation (ZNCC) [41, 119] between images a and b

$$C(a,b) = \frac{\sigma_{ab} + \epsilon}{\sigma_a \sigma_b + \epsilon} \tag{4.2}$$

where σ_{ab} is the covariance between a and b, and is defined as [119]

$$\sigma_{ab} = \frac{1}{m-1} \sum_{i=1}^{m} (a_i - \mu_a)(b_i - \mu_b) \tag{4.3}$$

m is the number of pixels in a and b, a_i and b_i are the ith pixel of a and b. μ_a, μ_b and σ_a, σ_b correspond to the mean intensity and standard deviations of a and b. ϵ is a constant to stabilize the division when $\sigma_a \sigma_b$ is close to zero. ZNCC a special case of structural similarity index (SSIM) and is less sensitive to the differences of illumination conditions and contrast of the two compared images [119]. The reader is referred to [130] for a comprehensive analysis on using SSIM and multi-scale SSIM as loss functions for neural networks and comparison experiments on these two loss functions based on the image restoration task.

Inspired by [23, 99], we introduce a segmentation consistency loss to further improve semantic consistency:

$$\begin{aligned}\mathcal{L}_{seg}(G,F) = \mathcal{L}_{CE}(x_m, S(F(G(x)))) \\ + \mathcal{L}_{CE}(S(y), S(G(F(y))))\end{aligned} \tag{4.4}$$

where $\mathcal{L}_{CE}(a, b)$ is the naive cross-entropy loss (Eq. (3.1)). S is a segmentation model trained on the labeled cadaveric dataset and its parameters are fixed during LC-GAN training. x_m is the ground truth segmentation mask of image x. For the segmentation consistency loss

between y and $G(F(y))$, we use the segmentation on the image y as the target mask because the labels of the live dataset are not available according to our assumption.

Finally, the overall objective function is given by

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y) + \mathcal{L}_{GAN}(F, D_X) \\ + \lambda_1 \mathcal{L}_{cyc}(G, F) + \lambda_2 \mathcal{L}_{SSim}(G, F) \\ + \lambda_3 \mathcal{L}_{seg}(G, F)\end{aligned} \tag{4.5}$$

where λ_i are hyper-parameters that balance the impact of the losses. The generators are trained to minimize the overall objective function and the discriminators are trained to maximize it.

4.2.4 Experiments and Results

Evaluation Metrics

The segmentation performance was evaluated based on Dice similarity coefficient (DSC) and Intersection over Union (IoU), which have been discussed in Section 3.3.2.

Implementation Details

We implemented image-to-image translation models using Tensorflow on a single Nvidia Tesla T4 GPU. The segmentation models were implemented on a 3.70GHz Intel i7-8700K CPU and two Nvidia GTX2080ti GPUs.

Datasets. We studied image translation between the cadaveric and live dataset of UW-Sinus-Surgery-C/L, which is described in Section 2. For the segmentation task on the live dataset, we separated live surgery images into a training set of 3504 frames from 2 videos and a test set of 1154 frames from the remaining 1 video. To speed up LC-GAN training, we cropped square image patches inscribed in the endoscopic area and downscaled these patches to 208×208. When trained LC-GAN, we found that to stabilize the model, we need to exclude images that have limited corresponding knowledge in the other domain.

Specifically, we excluded overexposed frames or frames with too strong specular reflections from the cadaveric dataset, and excluded images with electric cautery, which has the most different appearance with the instruments in the cadaveric dataset (see Figure 2.1 (n)), from the live datasets. 900 cadaveric surgery frames were selected for the X domain and 3174 live surgery frames were selected for the Y domain. Note that frames in the live test set for instrument segmentation were excluded in LC-GAN training.

LC-GAN Training. We trained LC-GAN with 6 epochs using the Adam optimizer. The learning rate was 0.00008 for the first half of training and then was linearly decayed to zero over the remaining epochs. Each batch consisted of one image from domain X and one image from domain Y. The hyper-parameters λ_1, λ_2 and λ_3 were set to 5, 1 and 2, respectively. For the structural similarity loss $\mathcal{L}_{SSim}$, we chose $n_s = 4$ and empirically set $\gamma_1 = 0.05$, $\gamma_2 = 0.33$, $\gamma_3 = 0.35$ and $\gamma_4 = 0.27$.

Compared image-to-image translation models. We compared LC-GAN with baseline and state-of-the-art image-to-image translation models include CycleGAN [133], UNIT [80] and MUNIT [51]. We first trained all three models until 250 epochs and chose the minimum epochs that made the models converge. We found CycleGAN stabilized after 200 epochs. UNIT and MUNIT failed to converge within 250 epochs. We then chose 9 epochs for these two models because after 9 epochs they became more unstable.

Results

Qualitative evaluation. Figure 4.4 shows examples of image-to-image translation results. We found that LC-GAN generally retained the shape and location of the instruments and the structures of sinus tissue in the resultant fake images. A majority of images in our live dataset have the instruments merged into the background due to strong reflection and blood. LC-GAN could successfully handle such cases most of the time as shown in Figure 4.4(a,b). In contrast, CycleGAN was less robust and might translate part of the background to an instrument as shown in Figure 4.4(a,c). Also, CycleGAN tended to increase the size of the instruments as shown in Figure 4.4(b,d). The other two comparison methods, UNIT and

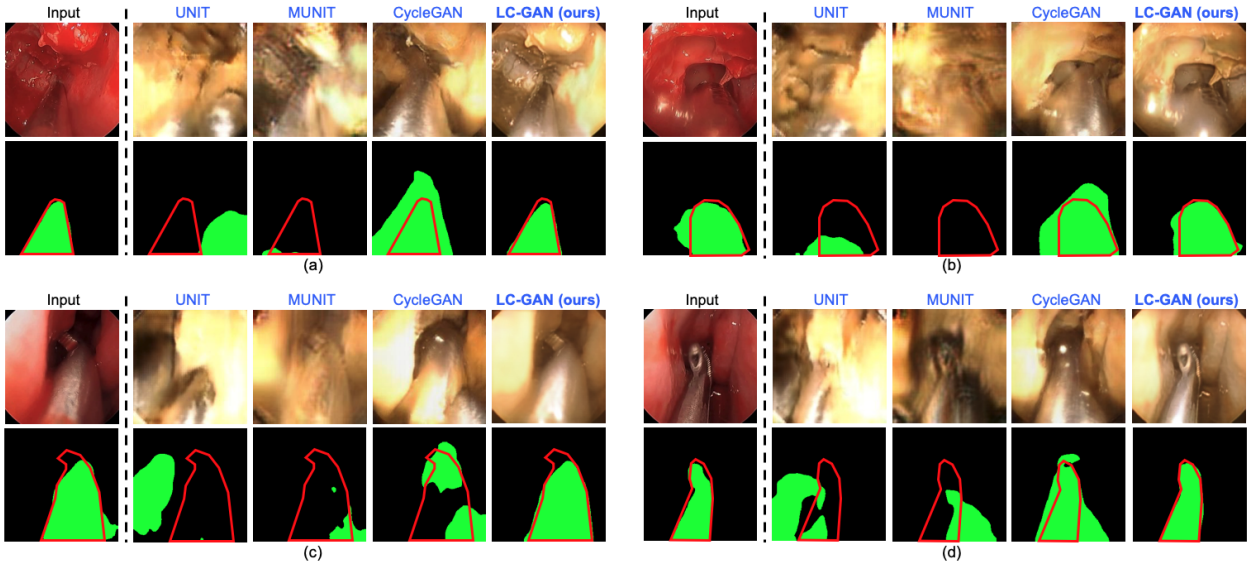

Figure 4.4: Examples of results from the proposed method and traditional method. In each subfigure, the last four columns of the top row show the fake-cadaveric images translated from the input real-live image using UNIT [80], MUNIT [51], CycleGAN [133] and LC-GAN (ours). The second row shows the corresponding instrument segmentation results obtained using DeepLabV3+ [18] with MAFA [97]. In the bottom row, the first segmentation is the result of the traditional method and the last four segmentations are from the proposed method. The predicted instrument regions are shown in green and the ground truth of the instrument contours are shown as red lines.

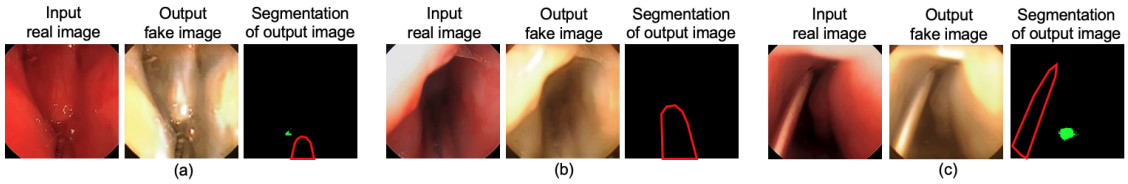

Figure 4.5: Failed live-to-cadaver translation examples given by LC-GAN. The corresponding instrument segmentation maps are obtained using DeepLabV3+ [18] with MAFA [97]. The predicted instrument regions are shown in green and the ground truth of the instrument contours are shown as red lines.

MUNIT, did not converge to the correct correspondence between the two image domains.

Although our image-to-image translation results are promising, the performance is still not satisfactory under challenging conditions. Figure 4.5 shows some typical fail cases. Figure

4.5(a) shows an example of an instrument merged in blood. Figure 4.5(b) shows an example of an instrument in shadow with a red appearance due to specular reflection. The instruments in Figure 4.5(a,b) are translated to regions similar to the surrounding background and lead to incorrect segmentations. Figure 4.5(c) represents another type of failure that happens when the instrument appears red and does not exist in the cadaveric dataset. In such a case, the color information is not reliable and the segmentation models should use shape information for successful segmentation. The segmentation models trained with the cadaveric dataset have not seen this or a similar instrument before, resulting in segmentation failure.

Quantitative evaluation. We used instrument segmentation results on the fake-cadaveric surgery images to quantitatively evaluate the image translation performance. The segmentation models were trained on the cadaveric dataset. We applied three segmentation models including DeepLabV3+ [18], TernausNet [53] and LWANet [90] with different pre-trained backbone feature extractors and a Multi-angle Feature Aggregation (MAFA) strategy [97]. The implementation details of the segmentation models are provided in [97]. As a comparison, we also performed the traditional method that trains and tests these segmentation models directly with labels of the live surgery dataset. The instrument segmentation performance was evaluated with DSC and IoU. Table 4.1 shows the segmentation results. We found that the segmentation results on fake-cadaveric surgery images translated by LC-GAN were better than the results obtained using other compared image-to-image translation models. Our method achieved 15.1%∼19.4% better mDSC and 17.9%∼22.9% better mIoU than using CycleGAN. Compared with the traditional method that uses the labeled live dataset, our method achieved 2.8%∼9.3% lower mDSC and 2.4%∼10.5% lower mIoU.

To evaluate the effectiveness of the proposed modules i) generator with trained backbone, ii) structural similarity loss $\mathcal{L}_{SSim}$ and iii) segmentation consistency loss $\mathcal{L}_{seg}$ in LC-GAN, we conducted ablation studies with different configurations. Table 4.2 shows the experiment results based on DeepLabV3+ [18]. All experiments were implemented with the same LC-GAN hyper-parameters described in Section 4.2.4. The network failed to converge without any proposed module in limited training epochs, while all three proposed modules helped

Table 4.1: Segmentation performances on live surgery dataset

Image-to-image Translation Model	Train Set	Test Set	Segmentation model (backbone), mDSC(%) / mIoU(%)			
			DeepLabV3+ (ResNet50)	TernausNet-16 (VGG16)	DeepLabV3+ (MobileNet)	LWANet (MobileNet)
UNIT [80]	Cadaver	Live	34.4/25.8	35.6/26.9	34.4/26.0	35.1/25.9
MUNIT [51]			22.4/17.8	18.9/14.2	23.8/18.1	20.9/15.3
CycleGAN [133]			62.6/51.4	59.9/48.7	59.6/48.2	57.7/46.2
LC-GAN (ours)			**79.9/73.1**	**75.1/68.1**	**79.0/71.1**	**72.8/64.1**
n/a	Live	Live	82.7/75.5	82.4/75.7	83.0/75.7	82.1/74.6

* All segmentation methods were implemented with MAFA [97]. The first four rows show the results of the UDA method, and the last row shows the results of the traditional method. The bold font indicates the best performance of the UDA method in each column.

Table 4.2: Ablation studies of LC-GAN with DeepLabV3+(ResNet50)

$\mathcal{L}_{SSim}$	$\mathcal{L}_{seg}$	Trained backbone	Segmentation peformance: mDSC(%) / mIoU(%)
×	×	×	24.1/16.8
√	×	×	78.2/71.4
×	√	×	77.7/70.9
×	×	√	78.2/71.0
√	√	×	79.0/72.2
√	×	√	78.4/71.6
×	√	√	79.1/72.5
√	√	√	**79.9/73.1**

* The bold font indicates the best performance.

the network converge faster and achieve acceptable performances. Moreover, combinations of two or three proposed modules led to 0.2%∼2.2% better mDSC and mIoU than using only one proposed module.

4.2.5 Discussions and Conclusions

We proposed an image-to-image translation model LC-GAN that achieved better semantic consistency using constraints that encourage structural similarity. The training of the proposed image-to-image translation model only requires an unpaired dataset, which can be easily extracted from the surgery videos.

Figure 4.4 shows that LC-GAN surpassed other comparison methods to provide the best image-to-image translation results. In contrast, CycleGAN tended to increase instrument size in the fake-cadaveric surgery images. This can be explained by the fact that in our dataset the instruments in the cadaveric domain are generally larger than the instruments in the live domain. UNIT and MUNIT failed to capture the correct mapping between the cadaveric and live surgery images. UNIT and MUNIT were built based on a shared-latent space assumption, i.e., each pair of corresponding images from the two domains can be mapped to a shared-latent space. However, this assumption may be too strict for our dataset because the scenes in the two domains have many differences in both instrument types and backgrounds.

Compared with the traditional method, LC-GAN achieved the lower performance. This result is as expected and the current gaps are acceptable. The live and cadaveric datasets have different types of instruments, so the distribution of the fake-cadaveric images is still different from the real-cadaveric images. Also, although we proposed two loss functions to improve the semantic consistency between the real images and their corresponding fake images, the semantic consistency has not been fully guaranteed. To mitigate the gap, we plan to create a small set of manually labeled fake-cadaveric images to fine-tune the deep segmentation models. Moreover, the proposed live surgery dataset introduces multiple challenges. When we labeled the dataset, we found that it is much easier to decide the instrument locations by referring to neighboring frames. This points out a future work direction, by extracting the temporal information of neighboring video frames, we can potentially obtain clues of semantic consistency for image-to-image translation. For example, we can extend the MFFA

module proposed in Chapter 3 to this work.

In this work, we proposed an image-to-image translation model LC-GAN to learn the mapping between two different but relevant image domains. We introduced structural similarity loss and segmentation consistency loss for LC-GAN to improve the semantic consistency during translation. We demonstrated the proposed model in a sinus surgery dataset of cadaveric and live surgery images. Our results show that the proposed method can potentially reduce the need to label more data for surgical instrument segmentation. These results have major implications on the ability to automatically segment and track surgical instruments, leading to improved analysis of surgery as well as enhancement in surgical training.

4.3 Unsupervised Domain Adaptation (UDA) with Feature Clustering-based Pseudo Labels

Many UDA methods align different domains without considering the semantic information of the data. For image classification, researchers have proposed to align domains by leveraging class information to improve adaptation performance. A common approach is to perform clustering along with task model training to reduce the domain gaps between samples of the same class [60, 43], as described in Section 4.1. In instrument segmentation, distinguishing instruments from the background can be very difficult due to the reflections, so aligning domains neglecting the semantic information may cause semantic inconsistency, which is discussed in Section 4.2.3. Inspired by the previous work on clustering-based domain alignment [60, 43], we propose to perform pixel feature clustering based on pixel class information and align domains by reducing the gaps between corresponding clusters of different domains.

4.3.1 Overview

Given a labeled dataset $S = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ in the source domain, where x_i is a surgical image and y_i is the corresponding ground truth segmentation map. Our goal is to train a model that can accurately segment the images of an unlabeled dataset $T = \{x_i^t\}_{i=1}^{N_t}$ in the target

domain.

The schematic of the proposed Clustering-based Transfer Network (CTN) is shown in Figure 4.6. The proposed method consists of two stages: First, the segmentation model is trained on datasets in the source domains. Then, the trained model is tuned on the target datasets with pseudo labels. In both stage, we combine a clustering method with the segmentation model to identify clusters of features output from the encoder. We perform clustering at two levels. The target task is binary segmentation that classifies each pixel as either instrument or background. Thus, the first level of clustering is to group the encoder features into an instrument cluster and a background cluster, as shown by the pink area and the blue area in Figure 4.6, respectively. Moreover, there is a high intra-class diversity in surgical images. Specifically, the background consists of different objects such as various tissues and fluid, bone, gauze, and other non-instrument objects, and an instrument may appear very different due to the strong reflection. Therefore, to cope with the high intra-class diversity, we further group the features of each cluster into sub-clusters as the second level of clustering. The sub-clusters are shown as circles in Figure 4.6. The CTN training involves

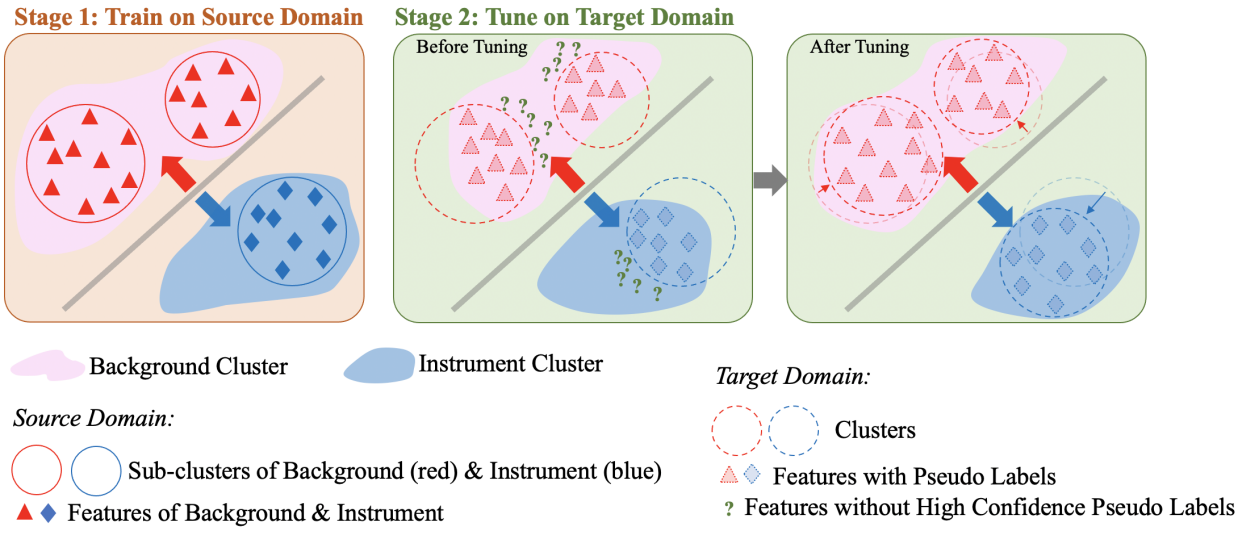

Figure 4.6: Overview of the CTN training procedure.

both levels of clustering. The details of CTN and its objective functions are described in Section 4.3.2. In the second stage, the target sub-cluster centers are initialized with the sub-cluster centers of the source data and are continue updated. At the beginning of each epoch, the pseudo labels are updated with the current model and the clustering information is used to estimate the confidence of the pseudo labels, which is further discussed in Section 4.3.3.

4.3.2 Clustering-based Transfer Network (CTN)

In stage 1, CTN is trained to perform segmentation and the first level clustering together. In stage 2, the segmentation model is initialized with the model trained in stage 1. Then we iteratively 1) update the sub-cluster centers of the encoder features, 2) update pseudo labels of the target data with the current model, and 3) train CTN with the pseudo labels for segmentation and provide better feature representation for the first level clustering. The training details of stage 2 are presented in Algorithm 1.

Objective functions for CTN. For segmentation, we calculate the naive cross-entropy loss $\mathcal{L}_{CE}$ using Eq. (3.1). During network training, we adopt Cauchy-Schwartz (CS) divergence [131] to estimate the first level feature clustering performance and guide the network to extract features that are well-separated in the instrument and background clusters. CS divergence reveals the distances between probability distributions and the CS divergence of two distributions is calculated as [57]

$$D_{CS} = -\log \frac{\int p_1(v)p_2(v)dv}{\sqrt{\int p_1^2(v)dv \int p_2^2(v)dv}} \tag{4.6}$$

where v is a feature output by the encoder, $p_1(\cdot)$ and $p_2(\cdot)$ are the probability density functions of the instrument and background clusters, respectively. By maximizing the CS divergence, the clusters tend to become more compact and well separated. According to [131], in practice, Eq. (4.6) can be calculated as

$$D_{CS} = \frac{\alpha_1^T G \alpha_2}{\sqrt{\alpha_1^T G \alpha_1 \alpha_2^T G \alpha_2}} \tag{4.7}$$

G is a distance matrix of encoder features, and each pairwise distance is calculated by $G_{m,n} = \exp(-\frac{d(v_m,v_n)}{2\sigma^2})$, where $d(\cdot)$ is the cosine dissimilarity [110] $d(a,b) = 1 - \cos(a,b) = 1 - \frac{a^T b}{\|a\|\|b\|}$. In stage 1, α_i is the i-th column of a cluster assignment matrix $A \in \mathcal{R}^{N_v \times 2}$ (N_v is the number of feature v), in which $A_{i,j} = 1$ indicates that the i-th feature v_i belongs to the j-th class. To obtain A, we first resize the ground truth segmentation map with nearest-neighbor interpolation to the same size as the feature maps, and then reshape the one-hot encoding of the resized ground truth map to $N_v \times 2$. In stage 2, because the ground truth segmentation maps are not available, we estimate a soft cluster assignment matrix A based on the sub-cluster centers. $A_{i,j}$ is first calculated as the summation of the cosine dissimilarities of the i-th feature to all sub-cluster centers of the j-th class, then each row of A is normalized with the softmax function.

Finally, for both stages, the overall objective function is giving by

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda D_{CS} \tag{4.8}$$

where λ is empirically chosen as 0.0001.

Encoder feature sub-clustering. Considering the high intra-class diversity in surgical images, we propose to further model each class with multiple sub-clusters. We apply spherical k-means clustering [14], which is a variant of k-means based on cosine dissimilarity, to estimate the sub-clusters. The network training in stage 1 does not involve sub-clusters. But at the end of stage 1, we perform clustering on both the instrument and background clusters to initialize sub-cluster centers for stage 2. The gap statistic method [113] is used to determine the sub-cluster number of each class. In stage 2, the sub-cluster centers are updated every n-th epoch based on the target data features. The choice of n is discussed in Section 4.3.4. We select encoder features that satisfy the following two conditions for sub-cluster updating: 1) The confidence of the feature's pseudo label is higher than the lower quartile of the confidences of all pseudo labels in the training set; 2) The feature is assigned to a sub-cluster that belongs to the same class as its pseudo label.

Algorithm 1: Stage 2- Tuning CTN on target dataset.

Input: Segmentation model trained on source dataset: M^s,
source sub-cluster centers: O^s,
target data: $\mathcal{T} = \{x_i^t\}_{i=1}^{N_t}$

Initialize target segmentation model M^t: $M^t \leftarrow M^s$;
Initialize target sub-cluster centers O^t: $O^t \leftarrow O^s$;
for *m epochs* **do**
    Update O^t every n-th epoch;
    Update pseudo labels:
        Assign pseudo labels to $\mathcal{T}$ using M^t;
        Assign each encoder feature to a sub-cluster using spherical k-means;
        Estimate pseudo label confidence using Eq. 4.9;
    for *n iterations* **do**
        Update network parameters θ by:
            $\theta \leftarrow \theta - \eta \cdot \bigtriangledown_\theta(\mathcal{L}_{CE} + \lambda D_{CS})$
    end
end

4.3.3 Pseudo Label

The schematic of generating pseudo labels and their corresponding confidence maps is shown in Figure 4.7. At the beginning of each epoch, the trained model is used to generate the temporary pseudo labels. In addition, we perform the second level clustering on the encoder output features and estimate the temporary confidence of the temporary pseudo label of each feature using Eq. (4.9). Next, the pseudo labels and their confidence maps used for model training in the current epoch are updated based on 1) the temporary pseudo labels and temporary confidence maps estimated at the beginning of the current epoch and 2) the pseudo labels and confidence maps used in the previous epoch.

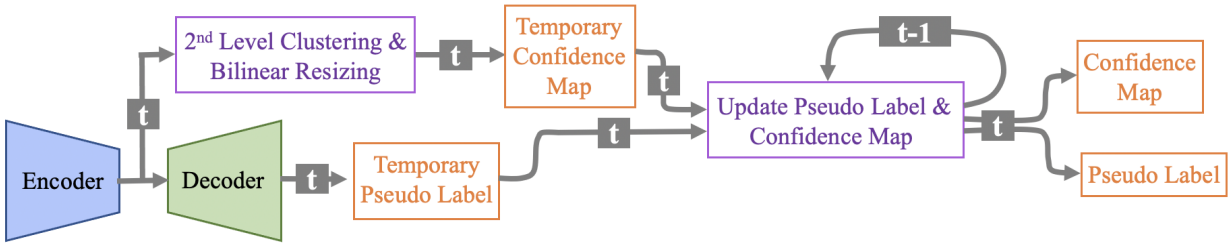

Figure 4.7: Schematic of generating pseudo labels and their corresponding confidence maps at the beginning of epoch t. The data, includes pseudo labels and confidence maps, are shown in orange boxes. The operations to generate or update pseudo labels and confidence maps are shown in purple boxes. The alphabet 't' or 't-1' on the arrows means the data flow is in the t-th or the $t-1$-th epoch, respectively.

Estimate temporary pseudo label confidence based on sub-clustering. The core of using pseudo labels for network training is to identify the accurate labels from noisy ones. In image segmentation, a common method is to generate pseudo labels with the current trained model and use the softmax output confidence to select data for further model training. For an input image with the size of $H \times W$, the decoder of a segmentation model usually outputs a confidence map $s \in \mathbb{R}^{H \times W \times C}$, where C is the total number of pixel classes. Then the softmax output is obtained by further normalizing this decoder output with the softmax function, and each element of the softmax output $s_{i,j,k}$ relates to the confidence of classifying the corresponding pixel $x_{i,j}$ to the k-th class. In this work, we also use the current model to obtain pseudo labels. In endoscopic sinus surgery, the instruments may have a similar appearance to the background due to reflection, so the models trained on the source domain may classify the instrument pixels of images in the target domain as background with high confidence. Therefore, selecting pseudo labels based on the softmax outputs may lead to suboptimal performance. We propose a feature clustering-based method to provide a more reliable confidence estimation. Because the encoder feature maps usually have a smaller size than the segmentation maps, so we first resize the temporary pseudo ground truth segmentation maps to the same size as the feature maps using nearest-neighbor interpolation

to assign each feature with a pseudo class label. Next, for a feature that has a pseudo label of class i, if its minimum distance to the sub-clusters of class i is much smaller than its distances to all sub-clusters that belong to all other classes, higher confidence should be assigned to this feature. For binary segmentation, assume the sub-cluster centers O consists of two groups of sub-clusters $C_1 = \{c_m^1\}_{m=1}^{k_1}$ and $C_2 = \{c_m^2\}_{m=1}^{k_2}$ (k_i is the number of sub-clusters of class i) for the instrument cluster and the background cluster, respectively. Then the label confidence of a encoder feature v classified as class i is calculated by

$$\sigma = 1 - \frac{\min\limits_{m} d(c_m^i, v)}{\min\limits_{m,\forall j \neq i} d(c_m^j, v)} \tag{4.9}$$

where $d(\cdot)$ calculates the cosine dissimilarity. Finally, the confidences of all features in the encoder feature maps form a confidence map, which is then further scaled to the same size as the segmentation map through bilinear resizing.

Update pseudo labels and confidence maps. To update the pseudo label and the corresponding confidence value of a pixel in a target image, we consider the following two situations:

- If the pixel belongs to the same class according to the current temporary pseudo label and the previous pseudo label, the pseudo-class label of this pixel will remain the same in the current epoch. The pixel's pseudo label confidence value will be the maximum of the corresponding confidence values in the current temporary confidence map and the previous confidence map.

- If the pixel belongs to different classes according to the current temporary pseudo label and the previous pseudo label, the pseudo-class label and the confidence of this pixel will be updated with the temporary pseudo-class label and the temporary confidence only when the temporary confidence is higher than the corresponding confidence in the previous confidence map. Otherwise, the pseudo-class label and the confidence value will remain the same in the current epoch.

4.3.4 Experiments and Results

We compared the proposed method with two domain adaptation methods for image segmentation. The first method CrDoCo [21] tries to align domains in the pixel level. The second method [114], denoted as 'Adapt Output' in Table 4.3, tries to align two domains in the feature level. The segmentation model used in the proposed and compared methods was DeepLabV3+ [18] with a backbone feature extractor MobileNet [48]. DeepLabV3+ was chosen because it has achieved state-of-the-art performance in many recent works on instrument segmentation [105, 97]. MobileNet was chosen because it is a representative lightweight backbone [90, 108], so using it allows all methods to be trained within a reasonable time, especially for CrDoCo that involves training a CycleGAN [133] and two segmentation models at the same time. In addition, the segmentation performance on the target dataset achieved with the segmentation model trained on the source dataset was compared as a baseline, which is denoted as 'Direct transfer' in Table 4.3.

Implementation Details

CTN and comparison domain adaptation methods were implemented on a 4.20GHz Intel i7-7700K CPU and a Nvidia Titan Xp GPU.

Datasets. We explored domain adaptation between the cadaveric and live dataset of UW-Sinus-Surgery-C/L (see Section 2). Specifically, we conducted experiments under the following two settings: 1) L→C: use the live dataset as the source dataset and the cadaveric dataset as the target dataset, 2) C→L: use the cadaveric dataset as the source dataset and the live dataset as the target dataset. For both experiment settings, we used all images from the source dataset to train the segmentation model in stage 1. Then in stage 2, the segmentation model was further trained and tested on the target domain based on the 3-fold cross-validation setting described in Section 2.

CTN Training. In stage 1, the segmentation model was trained using the Adam optimizer with 10k iterations and a batch size of 16. The learning rate was initialized as 0.0005 and

was exponentially decayed every 2500 iterations with a decay rate of 0.5. In stage 2, the segmentation model trained in stage 1 was further tuned with 3k iterations, a batch size of 16, and a learning rate of 0.0000625. In both stages, the images were augmented by 1) changing their hue, brightness, saturation and contrast and 2) flipping, rotation, scaling, and cropping.

Compared domain adaptation models. The two comparison domain adaptation methods [114, 21] and the segmentation model (DeepLabV3+ with MobileNet) for direct transfer were trained with 10k iterations and a batch size of 16. The learning rate was initialized as 0.0005 and was exponentially decayed every 2500 iterations with a decay rate of 0.5. The images were augmented in the same way as in CTN training.

Results

Table 4.3 shows the segmentation performance on the target dataset of the proposed and compared domain adaptation methods. The segmentation performance was evaluated based on Dice similarity coefficient (DSC) and Intersection over Union (IoU) (see Section 3.3.2). The performance of CrDoCo [21] was worse than direct transfer. This is because CrDoCo includes a CycleGAN model [133] and some of its loss functions are calculated based on the image translation results from CycleGAN, but stabilizing CycleGAN can be challenging as shown in Section 4.2.4. In contrast, 'Adapt Output' [114] and the proposed method achieved better performance than the baseline- direct transfer. The proposed method achieved 1.0%∼10.7% better mDice and 1.0%∼9.1% better mIoU than 'Adapt Output' when transferring knowledge from the live to the cadaveric dataset (L→C). But when the knowledge was transferred from the cadaveric to the live dataset (C→L), we did not see a significant difference between the performance of 'Adapt Output' and the proposed method.

We compared three different pseudo label confidence estimate methods and the results are shown in Table 4.4. Specifically, the pseudo label confidence are estimated based on: 1) 'Softmax': softmax output confidence, 2) 'Dist': the minimum distance between the feature (with a pseudo label of class i) and all sub-cluster centers of the i-th class, 3) 'Dist-ratio':

Table 4.3: Domain adaptation performance for segmentation task on UW-Sinus-Surgery-C/L

No.	Domain Adaptation Method	L→C	C→L
1	Direct transfer	60.5(2.8)/53.8(1.7)	30.9(4.9)/26.2(4.8)
2	CrDoCo [21]	15.8(6.6)/15.8(6.6)	8.2(3.0)/7.8(2.2)
	Adapt Output [114]	69.3(3.6)/62.2(3.3)	48.7(2.1)/**41.6(1.3)**
3	Ours	**76.4(1.5)**/**67.7(1.5)**	**52.4(1.7)**/40.7(1.9)

* The bold font indicates the best performance in the column.

the proposed distance ratio-based confidence (Eq. 4.9). Table 4.4 shows that CTN achieved the best performance based on the proposed confidence estimate method.

Table 4.4: Domain Adaptation Performance of CTN based on different pseudo label confidences on UW-Sinus-Surgery-C/L

Pseudo			L→C	C→L
Softmax	Dist	Dist-ratio		
✓			67.0(4.7)/58.9(4.8)	5.8(1.6)/5.8(1.6)
	✓		72.0(1.7)/62.3(1.3)	49.7(0.9)/**41.0(1.4)**
		✓	**76.4(1.5)**/**67.7(1.5)**	**52.4(1.7)**/40.7(1.9)

* The bold font indicates the best performance in the column.

4.3.5 Discussions and Conclusions

In this work, we explored a domain adaptation method for segmentation based on image feature clustering. The proposed method does not require the source and target data to be trained together. Thus, it can be applied to a situation when the source data can not be shared with the client, who wants to transfer the knowledge of a model trained on the source datasets to their target datasets, due to some privacy or security concerns.

Tables 4.3 and 4.4 show that the proposed CTN achieved notable improvement over the

direct transfer baseline. However, when we performed domain adaptation from the cadaveric to the live dataset, both the performance of CTN and 'Adapt Output' were still far from good enough. One reason that leads to the unsatisfactory performance on C→L is that the live dataset is more challenging than the cadaveric dataset because the blood and strong reflection of background on the instrument surface makes distinguishing the instrument from the background very difficult. Moreover, some instruments presented in the live dataset do not exist in the cadaveric dataset, such as the electric cautery. Therefore, during C→L, the target dataset (live dataset) includes knowledge that does not exist in the source dataset (cadaveric dataset), and the current framework of CTN has not included methods that allow the segmentation model to infer new knowledge by itself. Considering that in practical scenarios, it is common that one source dataset may not include all knowledge involved in the target dataset, we plan to explore multi-source domain adaptation to leverage knowledge from multiple source datasets in the future. The proposed feature clustering strategy could be further explored to select data that has more shared knowledge between different domains.

Chapter 5

OBJECTIVE SURGICAL SKILL ASSESSMENT

In the current surgeon training process, the students' performance is mainly evaluated by senior surgeons, which is subjective and time-consuming [16]. To reduce the subjectivity in evaluation, rating criteria such as Objective Structured Assessment of Technical Skill (OSATS) have been proposed [66, 86]. However, these skill assessment methods are still subjective and are not efficient enough for new surgical technologies such as minimally invasive surgery. Therefore, automatic and objective skill assessment methods are highly needed in medical training [40, 58]. In addition, the ability to analyze surgical skills could help robots or systems provide more appropriate assistance to surgeons during surgeries. In this Chapter, we present our initial studies on automatic surgical skill assessment.

5.1 Related Works on Automatic Objective Surgical Skill Assessment

Analyzing motion metrics of instrument trajectories is a common approach in previous objective skill assessment studies [93, 47, 109, 70, 37, 58, 40]. Section 5.2.3 provides an overview of commonly used motion metrics. More recently, a few methods have been proposed to extract features other than instrument movements from surgical videos for the skill assessment task or related tasks [29, 6, 111]. Ban *et al.* combined a LSTM model with a statistics model for surgical phase recognition [6]. Tanwani *et al.* proposed to learn visual representations of video frames in a semi-supervised fashion that pushes frames of different tasks away from each other in the embedding space [111]. Providing explicit guidance to medical students is more valuable than only giving the skill score. To generate guidance toward individual skill improvement, it is necessary to investigate technologies that could discover more information related to the surgical procedure. But, progress on objective surgical skill assessment

is relatively limited. The difficulty of this task itself and the limited availability of public datasets are two main reasons. To move forward, extending studies to real surgeries is one of the main challenges to be addressed. Existing studies mainly focus on dry lab training tasks such as suturing and knot tying [38]. There is little work performed on real surgery datasets [37, 58]. While many motion metrics have led to promising results in distinguishing surgical skill levels under dry lab settings, their effectiveness for evaluating real surgeries hasn't been well-explored.

5.2 Pilot Study on Surgical Skill Assessment for Endoscopic Sinus Surgery

5.2.1 Overview

In this work, we explore objective surgical skill assessment by analyzing surgical instrument trajectories. Specifically, we compute several motion metrics from the instrument trajectories and study the relationships between these motion metrics and surgical skill levels. The data studied in this work includes the instrument tip trajectories in the surgical videos, and instrument and endoscope trajectories in the 3D space collected by an optical surgical navigation system (see Section 2 for more details). To extract instrument trajectories from videos, we propose a tracking method based on the instrument segmentation results obtained in Section 3, as described in Section 5.2.2. Next, the involved motion metrics and the statistical analysis method to study the relationships between these motion metrics and the skill levels are introduced in Section 5.2.3.

5.2.2 Segmentation Map-based Instrument Tip Tracking

To obtain instrument tip trajectories in surgical videos, we first detect the instrument tip in each video frame and then apply a Kalman filter [26] to further smooth the instrument tip trajectories throughout the videos. We propose a geometric method to detect the instrument tip from the instrument segmentation map based on instrument shape information. Segmentation maps are images that indicate the instrument and background regions of the

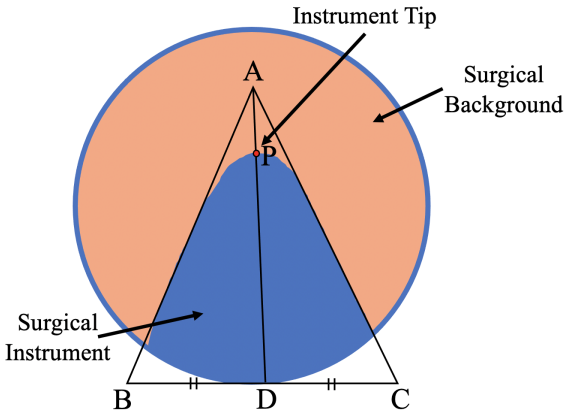

Figure 5.1: Schematic of the geometric method. The circular region is the endoscopic view on a video frame. $\triangle ABC$ is the enclosing triangle of the instrument region (blue). AD is the median that passes the triangle vertex A, which is the vertex closest to the endoscopic view center. The instrument tip P (red dot) is then identified as the intersection of the instrument contour and the median AD.

corresponding video frames (see Section 1.1) and are obtained using the method proposed in Chapter 3. While instrument tip tracking is not the focus of this work, we need to extract accurate instrument trajectories from videos to minimize the effect of tracking noise on skill assessment. To ensure the segmentation maps as accurate as possible, we train the segmentation model with all labeled frames in the target dataset and use the trained model to estimate the segmentation map of every frame in the videos.

Geometric method. The schematic of the geometric method is shown in Figure 5.1. We fit the enclosing triangle of the instrument region, and the tip is the intersection point of the instrument contour and the corresponding triangle median. For some challenging frames, the full instrument regions may not be successfully identified. In this case, the tip location estimated by the geometric method is unreliable. To reduce the trajectory noise, we only calculate the tip when the segmentation maps capture the general shape of instrument. Specifically, the instrument regions must satisfy two conditions: 1) the ratio of the instrument area and its convex hull area should be greater than 0.6; 2) at least one vertex of the enclosing triangle is inside the endoscopic view.

Kalman filter for tracking. The state-space representation of the instrument movement [26] is

$$s_{k+1} = \Phi_k s_k + w_k \tag{5.1}$$

$$z_k = H_k s_k + v_k \tag{5.2}$$

where $s_k = [x_k, y_k, \triangle x_k, \triangle y_k]^T$ is the state vector of the system, which consists of the position and velocity of the target; $z_k = [x_k, y_k]$ is the measurement vector of the target position; Φ_k is the state transition matrix and H_k is the measurement matrix; w_k is the vector of process noise, i.e. $E\{w_k w_k^T\} = Q_k$; v_k is the vector of measurement noise, i.e. $E\{v_k v_k^T\} = R_k$. In the target dataset UW-Sinus-Surgery-C, the instruments were handled by surgeons. Accurate dynamic system modeling for instrument movement is also a challenging task and is not within the scope of this study. To obtain a coarse approximate model, we adopt the assumption that the velocity is constant between two adjacent frames [27]. Therefore, Φ_k is defined as

$$\Phi_k = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.3}$$

H_k is defined as

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{5.4}$$

In addition, we also apply the Kalman filter to smooth the tracking data collected by the optical surgical navigation system. When the light of sight requirement is not satisfied, the optical tracking system cannot detect the location of the instrument or endoscope, which is then estimated using the Kalman filter.

5.2.3 Instrument and Endoscope Trajectory-based Skill Assessment

Motion metrics. The following motion metrics are explored in this work:

(1) *Total operative time* is the total time to perform the surgery [93, 47, 70].

(2) *Idle time* (2D&3D[1]) is the total time when the instrument is still [93]. In this work, we calculate the idle time from surgical videos and the 3D tracking data of the instrument using different methods. For the surgical videos, we estimate the idle time based on the corresponding instrument segmentation maps. We estimate instrument movement using the mDice [109] between instrument regions in consecutive frames and the larger mDice corresponds to smaller movement. We count idle time as when the mDice is greater than a threshold, which is empirically chosen as 0.98. For the instrument trajectories in the 3D space, we use the same method as previous work [93], i.e., the instrument was considered as still when its velocity is less than a certain threshold. The threshold is empirically chosen as 5mm.

(3) *Path length* (2D&3D) is the total distance traveled by the instrument or endoscope tip [93, 47, 37, 58].

(4) *Average velocity* (2D&3D) of instrument tip [93, 47, 70, 37].

(5) *Average acceleration* (2D&3D) of instrument tip [93].

(6) *Average smoothness* (2D&3D) is the average change in instrument tip acceleration [93, 47, 70].

(7) *Economy of volume* (3D) [93] estimates the ratio of space traveled by the instrument to the path length, i.e., it is calculated by

$$EoV = \frac{\sqrt[3]{[\max(X) - \min(X)] \cdot [\max(Y) - \min(Y)] \cdot [\max(Z) - \min(Z)]}}{\text{path length}} \tag{5.5}$$

(8) *Path length of relative movements between the instrument and endoscope* (3D).

To calculate these motion metrics, we first choose a data resampling rate and calculate the path length, average velocity, acceleration, and smoothness between adjacent data points. Then the total path length of a trajectory is calculated as the cumulative sum of the local path lengths. For the 3D trajectories, the economy of volume is further calculated based on the total path length. Also, the average velocity, acceleration, and smoothness throughout

[1]'2D&3D' indicates that the idle time is calculated on both surgical videos (2D) and 3D instrument trajectories. Metrics (3-8) are noted in the same way.

a trajectory are calculated as the average of the local average velocities, accelerations, and smoothnesses, respectively.

Statistical Analysis We perform statistical analysis to determine if a motion metric is related to surgical skill. The experiments are conducted on the UW-Sinus-Surgery-C dataset (see Section 2), in which the surgical performance presented in each video is labeled with the OSATS scores. The OSATS scores were generated by three expert surgeons reviewing the videos without the knowledge of the surgeons. There is a high diversity within scores given by different experts. For the total of 150 skill metrics (15 metrics for each video), the experts provided the same scores on only 22 metrics. On 81 metrics, two of the experts provided the same scores, and all experts gave different scores for the last 47 metrics. In this work, we average the three OSATS scores of each motion metric for the statistical analysis. A better method to combine the scores given by different experts should be explored in the future.

We calculate the Pearson's correlation coefficient between each motion metric and OSATS scores to evaluate if the metric can reveal or is related to the surgeons' skill levels. To determine the significant levels of these correlations, we perform multiple testing correction via false discovery rate (FDR) adjusted by Benjamini-Hochberg procedure [92].

5.3 Experiments and Results

5.3.1 Dataset.

We conducted experiments on the UW-Sinus-Surgery-C dataset, in which each video has OSATS scores generated by expert surgeons (see Section 2). The dataset consists of two types of data, 1) surgical videos (30fps) and 2) the tracking data of the endoscope and instrument collected by a surgical navigation system with a sampling rate of 10 Hz.

5.3.2 Statistical Analysis Results

We compared the correlation between the motion metrics and the corresponding OSATS scores. Since surgical phases are not studied in the current work, we only evaluated correla-

tions between motion metrics and the following 5 OSATS skill metrics: 1) Use of endoscopes; 2) Instrument handling; 3) Time and Motion; 4) Flow of Operation; 5) Overall surgical performance [66]. Table 5.1 shows the Pearson's correlation coefficients between motion metrics and OSATS scores. The motion metrics in Table 5.1 were calculated either from the surgical videos or the navigation-based tracking data, with a resampling rate of 3 Hz. The total operative time, all vision-based metrics, economy of volume of the 3D trajectories, and the path length estimated based on the 3D instrument trajectories had p-values less than 0.05 for their correlations with several OSATS scores, but none of the correlations were statistically significant according to FDR correction. Further, we assessed the sensitivity of the correlations between motion metrics and OSATS scores to the sampling rates. In Figure 5.2, we plotted the p-values of the correlations between motion metrics and OSATS overall surgical performance scores with test sampling rate from 1 to 5 Hz. The total operative time, path length, and economy of volume had stronger correlations with skill levels and were less sensitive to different sampling rates than other motion metrics. When using surgical videos, appropriate sampling rates that better reveal local velocity, acceleration, and smoothness led to stronger correlations between average velocity, acceleration and smoothness, and skill levels. In contrast, the relationships between the average velocity, acceleration, and smoothness of the 3D trajectories and the skill levels were weaker, no matter what sampling rate was chosen.

5.4 Discussions and Conclusions

In this work, we studied the relationship between surgical instrument movements and surgeons' skill levels in cadaveric endoscopic sinus surgery. The results provide some insights to guide future data collection and studies on objective skill assessment for real surgeries.

We calculated several motion metrics that have been validated in previous related work on dry lab datasets [93, 47, 70, 37, 58]. The motion metrics were extracted from both the instrument trajectories in the surgical videos and surgical navigation data in the 3D space. Table 5.1 and Figure 5.2 show that 1) total operative time, 2) idle time, 3) average

Table 5.1: Pearson's correlation coefficient between OSATS scores and automated motion metrics.

Motion Metrics	OSATS metrics, Pearson's correlation coefficient (p-value)				
	Use of Endoscopes	Instrument Handling	Time and Motion	Flow of Operation	Overall Performance
Total oper. time (1)	-0.78(0.008**)	-0.77(0.009**)	-0.86(0.002**)	-0.73(0.02*)	-0.73(0.02*)
Vision-based metrics					
Idle time (2)	-0.67(0.04*)	-0.80(0.005**)	-0.70(0.02*)	-0.65(0.04*)	-0.65(0.04*)
Path length (3)	-0.63(0.05*)	-0.70(0.03*)	-0.73(0.02*)	-0.67(0.03*)	-0.66(0.04*)
Ave. velocity (4)	0.73(0.02*)	0.65(0.04*)	0.75(0.01*)	0.62(0.06)	0.61(0.06)
Ave. acceleration (5)	0.76(0.01*)	0.67(0.03*)	0.79(0.007**)	0.67(0.04*)	0.65(0.04*)
Ave. smoothness (6)	0.78(0.008**)	0.68(0.03*)	0.80(0.006**)	0.69(0.03*)	0.67(0.03*)
Navigation-based metrics					
Endoscope					
Path length (3)	-0.43(0.21)	-0.58(0.08)	-0.63(0.05)	-0.47(0.17)	-0.49(0.15)
Ave. velocity (4)	0.30(0.40)	0.10(0.77)	0.13(0.73)	0.21(0.55)	0.19(0.59)
Ave. acceleration (5)	0.28(0.43)	0.14(0.69)	0.17(0.64)	0.24(0.51)	0.20(0.58)
Ave. smoothness (6)	0.31(0.38)	0.18(0.62)	0.20(0.58)	0.27(0.44)	0.24(0.50)
Economy of volume (7)	0.75(0.01*)	0.64(0.05*)	0.64(0.05*)	0.62(0.06)	0.67(0.03*)
Instrument					
Idle time (2)	-0.56(0.09)	-0.55(0.1)	-0.63(0.05)	-0.48(0.16)	-0.48(0.17)
Path length (3)	-0.54(0.11)	-0.61(0.06)	-0.65(0.04*)	-0.57(0.08)	-0.60(0.07)
Ave. velocity (4)	0.14(0.71)	0.09(0.80)	0.10(0.78)	0.08(0.82)	0.05(0.89)
Ave. acceleration (5)	0.11(0.77)	0.13(0.72)	0.14(0.69)	0.11(0.77)	0.06(0.88)
Ave. smoothness (6)	0.12(0.74)	0.14(0.69)	0.15(0.67)	0.13(0.73)	0.07(0.84)
Economy of volume (7)	0.65(0.04*)	0.49(0.15)	0.54(0.11)	0.56(0.09)	0.61(0.06)
Relative movements					
Path length (8)	-0.47(0.17)	-0.50(0.14)	-0.52(0.12)	-0.46(0.18)	-0.50(0.14)

* i) p-values < 0.01 are marked with '**', p-values < 0.05 are marked with '*'; ii) In the first column, the number after each motion metric is their ID in Section 5.2.3, where a detailed explanation of the metrics can be found.

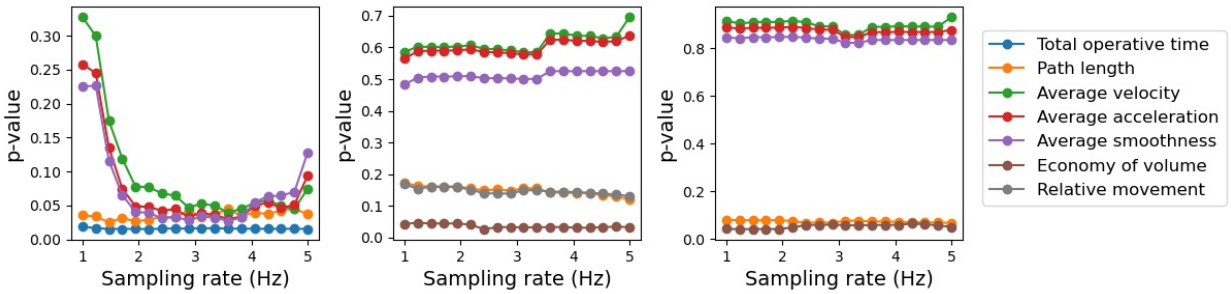

Figure 5.2: The p-values of the correlations between motion metrics and overall performance scores (OSATS) when using different sampling rates. *Left*: vision-based metrics. *Middle*: 3D endoscope trajectories-based metrics. *Right*: 3D instrument trajectories-based metrics.

velocity, acceleration, and smoothness of the instrument trajectories in the surgical videos, and 4) economy of volume of the surgical navigation data in the 3D space were related with skill levels to some degree. However, none of the motion metrics reached statistical significance according to the FDR-based multiple testing correction. One reason for this result is that the noise in instrument tracking data may adversely affect skill level analysis. When instrument trajectories were extracted from the surgical videos, although the proposed method is simple, its tip tracking performance was acceptable because the geometric method was performed based on accurate segmentation maps provided by the method proposed in Chapter 3. However, to investigate skill assessment on other datasets in the future, exploring more advanced vision-based tip tracking methods is necessary. For the navigation data, the Kalman filter was used to estimate the location of the instrument and endoscope when the tracking markers were occluded. However, in the studied surgeries, the tracking missing rate can be up to 40% for the instrument and up to 25 % for the endoscope. The approximate instrument movement model used in this work may not be enough to cope with such high tracking missing rate. To achieve more accurate tracking, methods for fusing multiple sensor data, such as combining the 3D tracking data with instrument segmentation results, should be considered in the future.

Moreover, our current dataset is not large enough (10 videos in total) for statistical

analysis and more surgical videos with skill levels labeled should be collected in the future. On the other hand, as discussed at the beginning of this Chapter, rating skill levels with OSATS is still subjective. Therefore, future work on objective skill assessment should not only rely on OSATS or other surgical skill evaluation rubrics to generate labels. Information such as surgical phases and treatment outcomes should also be collected and studied.

Chapter 6

CONCLUSIONS AND FUTURE WORK

This dissertation focuses on developing models that can leverage temporal information throughout surgical videos to improve segmentation, as well as exploring domain adaptation technologies to enhance the generalization ability of surgical instrument segmentation models. We also conducted initial studies on automatic and objective surgical skill assessment based on the segmentation results. To evaluate the proposed methods, an endoscopic sinus surgery dataset that consists of challenging, low-quality surgical images was proposed.

6.1 Temporal Information-based Segmentation Methods

We developed a model that aggregates video frame features temporally and spatially to improve instrument segmentation performance [72]. The process of passing and aggregating features throughout frame sequences can be approximately considered as distributing the computation load of deep feature extraction over sequential frames, which allows using lightweight encoders to reduce the computation costs. This work is among the first few studies that use the information to achieve more robust instrument segmentation. The proposed method directly concatenates feature maps of adjacent video frames without warping the previous feature maps to cope with the instrument movements between frames. As the instrument location differences are usually small in neighboring frames, this approach is acceptable for feature aggregation between most frames. However, to further improve the performance, methods that can accurately match the corresponding features between the previous and current feature maps to guide feature aggregation is necessary. Moreover, the other two topics explored in this work, i.e., domain adaptation and objective surgical skill assessment, could be benefit from better approach of extracting temporal information, which

will be investigated in the future.

6.2 Domain Adaptation for Instrument Segmentation

In this work, we proposed two unsupervised domain adaptation methods that transfer knowledge based on pixel-level and feature-level alignment, respectively. The first method LC-GAN focuses on improving image-to-image translation, which aims to map an image from one domain to images that have characteristics of another domain [76]. By translating images between different domains, the models trained on the domains that have labeled datasets can be leveraged to perform tasks on other domains without labels. The other method CTN tries to match the feature space of images from two domains based on feature clustering information [71]. The feature clusters of the unlabeled datasets are pushed toward their corresponding cluster centers of the labeled dataset to obtain pseudo labels for tuning models on the unlabeled data.

In the proposed live sinus surgery dataset, there are several types of instruments that do not exist in the cadaver dataset. Among these instruments, electric cautery has a very different appearance from the instruments used in the cadaver dataset. Considering this situation, when implementing LC-GAN, live images with electric cautery were excluded from the training phase as they make it more difficult for the translation models to converge. However, in real-world applications, it is common that a target dataset involves knowledge that does not exist in a source dataset, and such a need of selecting data with shared knowledge between domains will impede the application of this method. This partially motivated the development of CTN. The distances between an image feature and the feature cluster centers of a certain domain are related to the feature's similarity to this domain. In the future, we plan to explore using the clustering information to guide data selection for training the image translation models. On the other hand, both proposed methods were only implemented to transfer knowledge between pairs of datasets from different domains. As a source domain may not include all knowledge needed for performing tasks on a target domain, another future work will be exploring models that can efficiently acquire knowledge

from multiple source domains.

6.3 Objective Surgical Skill Assessment

We studied the relationships between several motion metrics of instrument trajectories and surgical skills presented during 10 cadaver surgeries [74, 73]. Although these motion metrics have been demonstrated for skill assessment on dry lab datasets, the statistical analysis results indicate that only using motion metrics may not be enough to evaluate skill during real surgeries, and more data should be collected to further evaluate them. Further, previous work mainly focused on classifying skill levels, while providing explicit suggestions to medical students and surgeons are highly needed. Our work focused on instrument segmentation due to time constraints, but we plan to explore if the segmentation results could be used to guide the deep models to extract more skill information from both the instrument and background regions of surgical videos in the future.

BIBLIOGRAPHY

[1] R Aggarwal, K Moorthy, and A Darzi. Laparoscopic skills training and assessment. *British Journal of Surgery*, 91(12):1549–1558, 2004.

[2] Max Allan, Satoshi Kondo, Sebastian Bodenstedt, Stefan Leger, Rahim Kadkhodamohammadi, Imanol Luengo, Felix Fuentes, Evangello Flouty, Ahmed Mohammed, Marius Pedersen, et al. 2018 robotic scene segmentation challenge. *arXiv preprint arXiv:2001.11190*, 2020.

[3] Max Allan, Alex Shvets, Thomas Kurmann, Zichen Zhang, Rahul Duggal, Yun-Hsuan Su, Nicola Rieke, Iro Laina, Niveditha Kalavakonda, Sebastian Bodenstedt, et al. 2017 robotic instrument segmentation challenge. *arXiv preprint arXiv:1902.06426*, 2019.

[4] Aleks Attanasio, Bruno Scaglioni, Elena De Momi, Paolo Fiorini, and Pietro Valdastri. Autonomy in surgical robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 4, 2020.

[5] Mohamed Attia, Mohammed Hossny, Saeid Nahavandi, and Hamed Asadi. Surgical tool segmentation using a hybrid deep CNN-RNN auto encoder-decoder. In *IEEE Int. Conf. Syst., Man, Cybern.*, pages 3373–3378, 2017.

[6] Yutong Ban, Guy Rosman, Thomas Ward, Daniel Hashimoto, Taisei Kondo, Hidekazu Iwaki, Ozanan Meireles, and Daniela Rus. Aggregating long-term context for learning laparoscopic and robot-assisted surgical workflows. *arXiv preprint arXiv:2009.00681*, 2020.

[7] Sebastian Bodenstedt, Max Allan, Anthony Agustinos, Xiaofei Du, Luis Garcia-Peraza-Herrera, Hannes Kenngott, Thomas Kurmann, Beat Müller-Stich, Sebastien Ourselin, Daniil Pakhomov, et al. Comparative evaluation of instrument segmentation and tracking methods in minimally invasive surgery. *arXiv preprint arXiv:1805.02475*, 2018.

[8] Sebastian Bodenstedt, Martin Wagner, Beat Peter Müller-Stich, Jürgen Weitz, and Stefanie Speidel. Artificial intelligence-assisted surgery: potential and challenges. *Visceral Medicine*, 36(6):450–455, 2020.

[9] Rositsa Bogdanova, Pierre Boulanger, and Bin Zheng. Depth perception of surgeons in minimally invasive surgery. *Surgical Innovation*, 23(5):515–524, 2016.

[10] David Bouget, Max Allan, Danail Stoyanov, and Pierre Jannin. Vision-based and marker-less surgical tool detection and tracking: a review of the literature. *Medical Image Analysis*, 35:633–654, 2017.

[11] David Bouget, Rodrigo Benenson, Mohamed Omran, Laurent Riffaud, Bernt Schiele, and Pierre Jannin. Detecting surgical tools by modelling local appearance and global shape. *IEEE Trans. Med. Imag.*, 34(12):2603–2617, 2015.

[12] Biagio Brattoli, Joseph Tighe, Fedor Zhdanov, Pietro Perona, and Krzysztof Chalupka. Rethinking zero-shot video classification: end-to-end training for realistic applications. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4613–4623, 2020.

[13] Daniel S Brown, Wonjoon Goo, and Scott Niekum. Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In *Conf. Robot Learn.*, pages 330–359. PMLR, 2020.

[14] Christian Buchta, Martin Kober, Ingo Feinerer, and Kurt Hornik. Spherical k-means clustering. *Journal of statistical software*, 50(10):1–22, 2012.

[15] Lien Calus, Nicholas Van Bruaene, Cedric Bosteels, Sarah Dejonckheere, Thibaut Van Zele, Gabrielle Holtappels, Claus Bachert, and Philippe Gevaert. Twelve-year follow-up study after endoscopic sinus surgery in patients with chronic rhinosinusitis with nasal polyposis. *Clinical and translational allergy*, 9(1):1–11, 2019.

[16] B Noland Carter. The fruition of Halsted's concept of surgical training. *Surgery*, 32(3):518–527, 1952.

[17] Abhishek Chaurasia and Eugenio Culurciello. LinkNet: exploiting encoder representations for efficient semantic segmentation. In *Proc. IEEE Visual Commun. Image Process.*, pages 1–4, 2017.

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

[19] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced LSTM for natural language inference. In *Proc. Annu. Meeting Assoc. Comput. Linguistics*, pages 1657–1668, 2017.

[20] Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun. No more discrimination: cross city adaptation of road scene segmenters. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1992–2001, 2017.

[21] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. CrDoCo: pixel-level domain transfer with cross-domain consistency. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1791–1800, 2019.

[22] Veronika Cheplygina, Marleen de Bruijne, and Josien PW Pluim. Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical image analysis*, 54:280–296, 2019.

[23] Anoop Cherian and Alan Sullivan. Sem-GAN: semantically-consistent image-to-image translation. In *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, pages 1797–1806, 2019.

[24] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[25] William Jay Conover and Ronald L Iman. On multiple-comparisons procedures. *Los Alamos Sci. Lab. Tech. Rep. LA-7677-MS*, 1:14, 1979.

[26] John L Crassidis and John L Junkins. *Optimal estimation of dynamic systems*. CRC press, 2011.

[27] Erik V Cuevas, Daniel Zaldivar, and Raul Rojas. Kalman filter for vision tracking. 2005.

[28] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: a large-scale hierarchical image database. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 248–255, 2009.

[29] Hazel Doughty, Dima Damen, and Walterio Mayol-Cuevas. Who's better? Who's best? Pairwise deep ranking for skill determination. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 6057–6066, 2018.

[30] Hao Shu Fang, Jianhua Sun, Runzhong Wang, Minghao Gou, Yong Lu Li, and Cewu Lu. InstaBoost: boosting instance segmentation via probability map guided copy-pasting. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 682–691, 2019.

[31] Evelyn Fix. *Discriminatory analysis: nonparametric discrimination, consistency properties*, volume 1. USAF school of Aviation Medicine, 1985.

[32] Adrian Ford and Alan Roberts. Colour space conversions. *Westminster University, London*, 1998:1–31, 1998.

[33] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937.

[34] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3146–3154, 2019.

[35] Isabel Funke, Sebastian Bodenstedt, Florian Oehme, Felix von Bechtolsheim, Jürgen Weitz, and Stefanie Speidel. Using 3D convolutional neural networks to learn spatiotemporal features for automatic surgical gesture recognition in video. In *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, pages 467–475. Springer, 2019.

[36] Isabel Funke, Sören Torge Mees, Jürgen Weitz, and Stefanie Speidel. Video-based surgical skill assessment using 3D convolutional neural networks. *Int. J. Comput. Assisted Radiol. Surgery*, 14(7):1217–1225, 2019.

[37] Sandeep Ganni, Sanne MBI Botden, Magdalena Chmarra, Richard HM Goossens, and Jack J Jakimowicz. A software-based tool for video motion tracking in the surgical skills assessment landscape. *Surgical endoscopy*, 32(6):2994–2999, 2018.

[38] Yixin Gao, S Swaroop Vedula, Carol E Reiley, Narges Ahmidi, Balakrishnan Varadarajan, Henry C Lin, Lingling Tao, Luca Zappella, Benjamın Béjar, David D Yuh, et al. JHU-ISI gesture and skill assessment working set (JIGSAWS): a surgical activity dataset for human motion modeling. In *MICCAI workshop: M2cai*, volume 3, page 3, 2014.

[39] Luis C García-Peraza-Herrera, Wenqi Li, Lucas Fidon, Caspar Gruijthuijsen, Alain Devreker, George Attilakos, Jan Deprest, Emmanuel Vander Poorten, Danail Stoyanov, Tom Vercauteren, et al. ToolNet: holistically-nested real-time segmentation of robotic surgical tools. In *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, pages 5717–5722, 2017.

[40] Benjamin Gautier, Harun Tugal, Benjie Tang, Ghulam Nabi, and Mustafa Suphi Erden. Laparoscopy instrument tracking for single view camera and skill assessment. In *Proc. Int. Conf. Robot. Autom.*, pages 5039–5045, 2019.

[41] Andrea Giachetti. Matching techniques to compute image motion. *Image and Vision Computing*, 18(3):247–260, 2000.

[42] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances Neural Inf. Process. Syst.*, pages 2672–2680, 2014.

[43] Xiang Gu, Jian Sun, and Zongben Xu. Spherical space domain adaptation with robust pseudo-label loss. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 9101–9110, 2020.

[44] Ralitza Gueorguieva and John H Krystal. Move over ANOVA: progress in analyzing repeated-measures data andits reflection in papers published in the archives of general psychiatry. *Archives of General Psychiatry*, 61(3):310–317, 2004.

[45] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 4918–4927, 2019.

[46] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 770–778, 2016.

[47] Erlend Fagertun Hofstad, Cecilie Våpenstad, Magdalena Karolina Chmarra, Thomas Langø, Esther Kuhry, and Ronald Mårvik. A study of psychomotor skills in minimally invasive surgery: what differentiates expert and nonexpert performance. *Surgical endoscopy*, 27(3):854–863, 2013.

[48] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[49] Ping Hu, Fabian Caba, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. Temporally distributed networks for fast video semantic segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 8818–8827, 2020.

[50] Jiang Hua, Liangcai Zeng, Gongfa Li, and Zhaojie Ju. Learning for a robot: deep reinforcement learning, imitation learning, transfer learning. *Sensors*, 21(4):1278, 2021.

[51] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proc. Europ. Conf. Comput. Vis.*, pages 172–189, 2018.

[52] Tae Hyun Kim, Kyoung Mu Lee, Bernhard Scholkopf, and Michael Hirsch. Online video deblurring via dynamic temporal blending network. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 4038–4047, 2017.

[53] Vladimir Iglovikov and Alexey Shvets. TernausNet: U-Net with VGG11 encoder pre-trained on imagenet for image segmentation. *arXiv preprint arXiv:1801.05746*, 2018.

[54] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 5070–5079, 2019.

[55] Mobarakol Islam, Yueyuan Li, and Hongliang Ren. Learning where to look while tracking instruments in robot-assisted surgery. In *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, pages 412–420, 2019.

[56] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1125–1134, 2017.

[57] Robert Jenssen, Jose C Principe, Deniz Erdogmus, and Torbjørn Eltoft. The Cauchy-Schwarz divergence and Parzen windowing: connections to graph theory and mercer kernels. *Journal of the Franklin Institute*, 343(6):614–629, 2006.

[58] Amy Jin, Serena Yeung, Jeffrey Jopling, Jonathan Krause, Dan Azagury, Arnold Milstein, and Li Fei-Fei. Tool detection and operative skill assessment in surgical videos using region-based convolutional neural networks. In *IEEE Winter Conf. Appl. Comput. Vis.*, pages 691–699, 2018.

[59] Yueming Jin, Keyun Cheng, Qi Dou, and Pheng-Ann Heng. Incorporating temporal prior from motion flow for instrument segmentation in minimally invasive surgery video. In *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, pages 440–448, 2019.

[60] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4893–4902, 2019.

[61] David W Kennedy and Brent A Senior. Endoscopic sinus surgery: a review. *Otolaryngologic Clinics of North America*, 30(3):313–330, 1997.

[62] A Khanna and A Sama. Managing complications and revisions in sinus surgery. *Current Otorhinolaryngology Reports*, 7(1):79–86, 2019.

[63] Diederik P Kingma and Jimmy Ba. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[64] Wouter M Kouw and Marco Loog. An introduction to domain adaptation and transfer learning. *arXiv preprint arXiv:1812.11806*, 2018.

[65] Helena Chmura Kraemer and Christine Blasey. *How many subjects?: statistical power analysis in research*. Sage Publications, 2015.

[66] Kulsoom Laeeq, Scott Infusino, Sandra Y Lin, Douglas D Reh, Masaru Ishii, Jean Kim, Andrew P Lane, and Nasir I Bhatti. Video-based assessment of operative competency in endoscopic sinus surgery. *American journal of rhinology & allergy*, 24(3):234–237, 2010.

[67] Simon Leonard, Ayushi Sinha, Austin Reiter, Masaru Ishii, Gary L Gallia, Russell H Taylor, and Gregory D Hager. Evaluation and stability analysis of video-based navigation system for functional endoscopic sinus surgery on in vivo clinical data. *IEEE Trans. Med. Imag.*, 37(10):2185–2195, 2018.

[68] Yuexiang Li, Jiawei Chen, Xinpeng Xie, Kai Ma, and Yefeng Zheng. Self-loop uncertainty: a novel pseudo-label for semi-supervised medical image segmentation. In *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, pages 614–623. Springer, 2020.

[69] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 6936–6945, 2019.

[70] Ke Liang, Yuan Xing, Jianmin Li, Shuxin Wang, Aimin Li, and Jinhua Li. Motion control skill assessment based on kinematic analysis of robotic end-effector movements. *Int. J. Med. Robot. Computer Assisted Surgery*, 14(1):e1845, 2018.

[71] (In preparation) Shan Lin, Fangbo Qin, Randall A Bly, Kris S Moe, and Blake Hannaford. Clustering-based unsupervised domain adaptation for surgical instrument segmentation. 2021.

[72] (Revision Submitted to IEEE Robot. Autom. Lett.) Shan Lin, Fangbo Qin, Haonan Peng, Randall A Bly, Kris S Moe, and Blake Hannaford. Multi-frame feature aggregation for real-time instrument segmentation in endoscopic video. *arXiv preprint arXiv:2011.08752*, 2020.

[73] Shan Lin, Xinyu Gu, Randall A Bly, Kris S Moe, and Blake Hannaford. Video-based automatic and objective endoscopic sinus surgery skill assessment. In *Medical Imaging: Image-Guided Procedures, Robotic Interventions, and Modeling*, volume 11315, page 113152L, 2020.

[74] Shan Lin, Fangbo Qin, Randall A Bly, Kris S Moe, and Blake Hannaford. Automatic sinus surgery skill assessment based on instrument segmentation and tracking in endoscopic video. In *Int. Workshop Multiscale Multimodal Med. Imag.*, pages 93–100, 2019.

[75] Shan Lin, Fangbo Qin, Randall A Bly, Kris S Moe, and Blake Hannaford. UW sinus surgery cadaver/live dataset (UW-Sinus-Surgery-C/L). 2020.

[76] Shan Lin, Fangbo Qin, Yangming Li, Randall A Bly, Kris S Moe, and Blake Hannaford. LC-GAN: image-to-image translation based on generative adversarial network for endoscopic images. pages 2914–2920, 2020.

[77] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 2117–2125, 2017.

[78] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

[79] Gang Liu and Jiabao Guo. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing*, 337:325–338, 2019.

[80] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances Neural Inf. Process. Syst.*, pages 700–708, 2017.

[81] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3431–3440, 2015.

[82] Yi Lu, Yaran Chen, Dongbin Zhao, Bao Liu, Zhichao Lai, and Jianxin Chen. CNN-G: convolutional neural network combined with graph for image segmentation with theoretical analysis. *IEEE Trans. Cogn. Devel. Syst.*, 2020.

[83] Michael J Mack. Minimally invasive and robotic surgery. *Jama*, 285(5):568–572, 2001.

[84] Faisal Mahmood, Richard Chen, and Nicholas J Durr. Unsupervised reverse domain adaptation for synthetic medical images via adversarial training. *IEEE Trans. Med. Imag.*, 37(12):2572–2581, 2018.

[85] Lena Maier-Hein, Martin Wagner, Tobias Ross, Annika Reinke, Sebastian Bodenstedt, Peter M Full, Hellena Hempe, Diana Mindroc-Filimon, Patrick Scholz, Thuy Nuong Tran, et al. Heidelberg colorectal data set for surgical data science in the sensor operating room. *Scientific data*, 8(1):1–11, 2021.

[86] JA Martin, Glenn Regehr, Richard Reznick, Helen Macrae, John Murnaghan, Carol Hutchison, and M Brown. Objective structured assessment of technical skill (OSATS) for surgical residents. *British journal of surgery*, 84(2):273–278, 1997.

[87] Jiaxu Miao, Yunchao Wei, and Yi Yang. Memory aggregation networks for efficient interactive video object segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 10366–10375, 2020.

[88] Debraj Mukherjee, Hasan A Zaidi, Thomas Kosztowski, Kaisorn L Chaichana, Henry Brem, and David C Chang. Disparities in access to neuro-oncologic care in the united states. *Archives of surgery*, 145(3):247–253, 2010.

[89] Seungjun Nah, Sanghyun Son, and Kyoung Mu Lee. Recurrent neural networks with intra-frame iterations for video deblurring. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 8102–8111, 2019.

[90] Zhen Liang Ni, Gui Bin Bian, Zeng Guang Hou, Xiao Hu Zhou, Xiao Liang Xie, and Zhen Li. Attention-guided lightweight network for real-time segmentation of robotic surgical instruments. In *Proc. Int. Conf. Robot. Autom.*, pages 9939–9945, 2020.

[91] Zhen-Liang Ni, Gui-Bin Bian, Guan-An Wang, Xiao-Hu Zhou, Zeng-Guang Hou, Hua-Bin Chen, and Xiao-Liang Xie. Pyramid attention aggregation network for semantic segmentation of surgical instruments. In *Proc. AAAI Conf. Artif. Intell.*, volume 34, pages 11782–11790, 2020.

[92] William S Noble. How does multiple testing correction work? *Nature biotechnology*, 27(12):1135, 2009.

[93] Ignacio Oropesa, Patricia Sánchez-González, Magdalena K Chmarra, Pablo Lamata, Alvaro Fernández, Juan A Sánchez-Margallo, Frank Willem Jansen, Jenny Dankelman, Francisco M Sánchez-Margallo, and Enrique J Gómez. EVA: laparoscopic instrument tracking based on endoscopic video analysis for psychomotor skills assessment. *Surgical endoscopy*, 27(3):1029–1039, 2013.

[94] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2009.

[95] Micha Pfeiffer, Isabel Funke, Maria R Robu, Sebastian Bodenstedt, Leon Strenger, Sandy Engelhardt, Tobias Roß, Matthew J Clarkson, Kurinchi Gurusamy, Brian R Davidson, et al. Generating large labeled data sets for laparoscopic image processing tasks using unpaired image-to-image translation. In *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, pages 119–127, 2019.

[96] Fangbo Qin, Yangming Li, Yun-Hsuan Su, De Xu, and Blake Hannaford. Surgical instrument segmentation for endoscopic vision with data fusion of CNN prediction and kinematic pose. In *Proc. Int. Conf. Robot. Autom.*, pages 9821–9827, 2019.

[97] Fangbo Qin, Shan Lin, Yangming Li, Randall A Bly, Kris S Moe, and Blake Hannaford. Towards better surgical instrument segmentation in endoscopic vision: multi-angle feature aggregation and contour supervision. *IEEE Robot. Autom. Lett.*, 5(4):6639–6646, 2020.

[98] Yidan Qin, Seyedshams Feyzabadi, Max Allan, Joel W Burdick, and Mahdi Azizian. daVinciNet: joint prediction of motion and surgical state in robot-assisted surgery. *arXiv preprint arXiv:2009.11937*, 2020.

[99] Pierluigi Zama Ramirez, Alessio Tonioni, and Luigi Di Stefano. Exploiting semantics in adversarial training for image-level domain adaptation. In *Proc. IEEE Int. Conf Image Process. Appl. Syst.*, pages 49–54, 2018.

[100] Carol E Reiley, Henry C Lin, David D Yuh, and Gregory D Hager. Review of methods for objective surgical skill evaluation. *Surgical endoscopy*, 25(2):356–366, 2011.

[101] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: an uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329*, 2021.

[102] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: convolutional networks for biomedical image segmentation. In *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, pages 234–241, 2015.

[103] Jacob Rosen, Blake Hannaford, Mark P MacFarlane, and Mika N Sinanan. Force controlled and teleoperated endoscopic grasper for minimally invasive surgery-experimental performance evaluation. *IEEE Trans. Biomed. Eng.*, 46(10):1212–1221, 1999.

[104] Tobias Ross, Annika Reinke, Peter M Full, Martin Wagner, Hannes Kenngott, Martin Apitz, Hellena Hempe, Diana Mindroc Filimon, Patrick Scholz, Thuy Nuong Tran, et al. Robust medical instrument segmentation challenge 2019. *arXiv preprint arXiv:2003.10299*, 2020.

[105] Tobias Roß, Annika Reinke, Peter M Full, Martin Wagner, Hannes Kenngott, Martin Apitz, Hellena Hempe, Diana Mindroc-Filimon, Patrick Scholz, Thuy Nuong Tran, et al. Comparative validation of multi-instance instrument segmentation in endoscopy: results of the ROBUST-MIS 2019 challenge. *Medical image analysis*, 70:101920, 2021.

[106] Alexey A Shvets, Alexander Rakhlin, Alexandr A Kalinin, and Vladimir I Iglovikov. Automatic instrument segmentation in robot-assisted surgery using deep learning. In *Proc. IEEE Int. Conf. Mach. Learn. Appl.*, pages 624–628, 2018.

[107] Yun-Hsuan Su, Kevin Huang, and Blake Hannaford. Multicamera 3D reconstruction of dynamic surgical cavities: camera grouping and pair sequencing. In *Proc. Int. Symp. Med. Robot.*, pages 1–7, 2019.

[108] Yanwen Sun, Bo Pan, and Yili Fu. Lightweight deep neural network for real-time instrument semantic segmentation in robot assisted minimally invasive surgery. *IEEE Robot. Autom. Lett.*, 6(2):3870–3877, 2021.

[109] Abdel Aziz Taha and Allan Hanbury. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC Medical Imaging*, 15(1):29, 2015.

[110] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. Data mining introduction, 2006.

[111] Ajay Kumar Tanwani, Pierre Sermanet, Andy Yan, Raghav Anand, Mariano Phielipp, and Ken Goldberg. Motion2Vec: semi-supervised representation learning from surgical videos. In *Proc. Int. Conf. Robot. Autom.*, pages 2174–2181, 2020.

[112] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of graphics tools*, 9(1):23–34, 2004.

[113] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[114] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 7472–7481, 2018.

[115] John Wilder Tukey. The collected works of John W. Tukey. 1, 1984.

[116] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[117] Jean-Philippe Vert, Koji Tsuda, and Bernhard Schölkopf. A primer on kernel methods. *Kernel methods in computational biology*, 47:35–70, 2004.

[118] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. FEELVOS: fast end-to-end embedding learning for video object segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 9481–9490, 2019.

[119] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.

[120] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *Asilomar Conf. Signals Syst. Comput.*, volume 2, pages 1398–1402, 2003.

[121] Kai Xu and Fengbo Ren. CSVideoNet: a real-time end-to-end learning framework for high-frame-rate video compressive sensing. In *IEEE Winter Conf. Appl. Comput. Vis.*, pages 1680–1688. IEEE, 2018.

[122] Xin Yang, Lequan Yu, Shengli Li, Xu Wang, Na Wang, Jing Qin, Dong Ni, and Pheng-Ann Heng. Towards automatic semantic segmentation in volumetric ultrasound. In *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, pages 711–719. Springer, 2017.

[123] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proc. Int. Joint Conf. Artif. Intell.*, pages 3634–3640, 2018.

[124] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation*, 31(7):1235–1270, 2019.

[125] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S^4L: self-supervised semi-supervised learning. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1476–1485, 2019.

[126] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. CANet: class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 5217–5226, 2019.

[127] Yifan Zhang, Lei Shi, Yi Wu, Ke Cheng, Jian Cheng, and Hanqing Lu. Gesture recognition based on deep deformable 3D convolutional neural networks. *Pattern Recognition*, 107:107416, 2020.

[128] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. HSA-RNN: hierarchical structure-adaptive RNN for video summarization. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 7405–7414, 2018.

[129] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. CAM-RNN: co-attention model based RNN for video captioning. *IEEE Trans. Image Process.*, 28(11):5552–5565, 2019.

[130] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Trans. Comput. Imag.*, 3(1):47–57, 2016.

[131] Runwu Zhou and Yi-Dong Shen. End-to-end adversarial-attention network for multi-modal clustering. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 14619–14628, 2020.

[132] Wei Zhou, Yukang Wang, Jiajia Chu, Jiehua Yang, Xiang Bai, and Yongchao Xu. Affinity space adaptation for semantic segmentation across domains. *IEEE Trans. Image Process.*, 2020.

[133] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 2223–2232, 2017.

[134] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proc. IEEE*, 109(1):43–76, 2020.