Multi-frame Feature Aggregation for Real-time Instrument Segmentation in Endoscopic Video

Shan Lin\textsuperscript{1}, Fangbo Qin\textsuperscript{2}, Haonan Peng\textsuperscript{1}, Randall A. Bly\textsuperscript{1}, Kris S. Moe\textsuperscript{1}, and Blake Hannaford\textsuperscript{1}, Fellow, IEEE

Abstract—Deep learning-based methods have achieved promising results on surgical instrument segmentation. However, the high computation cost may limit the application of deep models to time-sensitive tasks such as online surgical video analysis for robotic-assisted surgery. Moreover, current methods may still suffer from challenging conditions in surgical images such as various lighting conditions and the presence of blood. We propose a novel Multi-frame Feature Aggregation (MFFA) module to aggregate video frame features temporally and spatially in a recurrent mode. By distributing the computation load of deep feature extraction over sequential frames, we can use a lightweight encoder to reduce the computation costs at each time step. Moreover, public surgical videos usually are not labeled frame by frame, so we develop a method that can randomly synthesize a surgical frame sequence from a single labeled frame to assist network training. We demonstrate that our approach achieves superior performance to corresponding deeper segmentation models on two public surgery datasets.

Index Terms—Computer vision for medical robotics, deep learning for visual perception, object detection, segmentation and categorization.

I. INTRODUCTION

SURGICAL videos provide information that surgeons rely on during operations. As video-assisted and robotic-assisted surgery become more integrated, analyzing surgical videos becomes more essential. Surgical video instrument segmentation is one important task that can provide instrument locations to robotic and computer-assisted surgery systems\textsuperscript{1},\textsuperscript{2},\textsuperscript{3}. However, challenging conditions in surgical videos such as various lighting scenarios and blood make instrument segmentation a hard problem. For some challenging data, identifying the instrument regions without looking at adjacent frames can be hard even for human eyes\textsuperscript{4}, so algorithms that could leverage the temporal information for segmentation are attractive. Although Convolutional Neural Networks (CNN) have been successfully applied in instrument segmentation\textsuperscript{3},\textsuperscript{5},\textsuperscript{6},\textsuperscript{7}, existing works usually perform segmentation on a single frame without considering the temporal information.

Recurrent Neural Networks (RNNs) are typical models for temporal sequences and have been successfully applied in robotic and computer-assisted surgery systems\textsuperscript{1},\textsuperscript{2},\textsuperscript{3}.\textsuperscript{1} This work is recommended for publication by Editor Eric Marchand upon evaluation of the Associate Editor and Reviewers’ comments.\textsuperscript{1}Shan Lin, Haonan Peng, and Blake Hannaford are with Department of Electrical & Computer Engineering, University of Washington (UW), Seattle, WA 98195, USA. Randall A. Bly and Kris S. Moe are with Department of Otolaryngology-Head & Neck Surgery, UW, Seattle, USA. shan13@uw.edu\textsuperscript{1}Fangbo Qin is with Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. qinfangbo2013@ia.ac.cn

Digital Object Identifier (DOI): see top of this page.
Ni et al. used relationships between feature channels to emphasize instrument regions for better segmentation [15]. Additionally, Jin et al. proposed to propagate the instrument region detected in the previous frame using motion flow as prior information to assist instrument segmentation [16]. Although initial works have shown promising results, studies that leverage temporal and spatial information for instrument segmentation are still limited.

General Video Analysis Methods. As typical networks that can model temporal behavior, RNNs have achieved state-of-the-art performance in many video analysis tasks such as video captioning, summarization, deblurring [17], [18]. In contrast, applications of RNNs in video semantic segmentation are limited due to difficulties in training RNNs. In addition, 3D CNNs have been applied for video analysis, including gesture recognition and video classification [20], [21], [22]. Although achieving promising performance with their rich spatiotemporal feature representation, 3D CNNs usually have more parameters and higher computation costs than 2D CNNs. Aggregating neighboring frame features has been proposed as an alternative approach to leverage temporal information for segmentation [12]. Another approach is to propagate the segmentation of the previous frame to assist segmentation on the current frame [23], [24].

III. METHODS

We develop a Multi-frame Feature Aggregation (MFFA) module that aggregates features both temporally and spatially for segmentation. MFFA is designed to be flexibly combined with general encoder-decoder segmentation models as introduced in Section II-C. The architecture of MFFA is presented in Section III-A.

In many public datasets for surgical instrument segmentation [13], [4], frames are extracted 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 perform an initial investigation on using synthetic sequences with every frame labeled for training, as described in Section III-D. Each synthetic sequence consists of a real labeled frame and several frames synthesized based on this real frame.

A. Task Description and Overall Framework

Given an input sequence with $N$ video frames $X = \{x_1, x_2, ..., x_N\}$, where $x_i$ is the $i$th frame in the sequence. The task is to predict the corresponding binary masks $Y = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_N\}$, where $\hat{y}_i$ indicates the predicted instrument and background regions of $x_i$. For some $x_i$, the corresponding ground truth label $y_i$ is available for training or testing.

The proposed framework is shown in Fig. 1. Instead of segmenting each frame separately, we propose MFFA to aggregate features from the previous frame and pass the aggregated 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. After obtaining the feature maps $f_i$, outputted from the encoder, $f_{i+1}$, the previous output of MFFA $h_{i-1}$, and the previous predicted segmentation mask $\hat{y}_{i-1}$ are inputted into MFFA to generate aggregated feature maps $h_i$. $h_i$ is then passed to the next frame and inputted to the decoder for segmenting the current frame. Finally, the encoder generates a softmax output $s_i$. The segmentation mask $\hat{y}_i$ is then calculated with argmax from $s_i$.

Since MFFA is designed for general segmentation models, we demonstrate the proposed method based on a popular segmentation model DeepLabV3+ [25] with two representative backbone feature extractors, ResNet50 [26] and MobileNet [27]. DeepLabV3+ and the two backbones have achieved state-of-the-art performance in many recent works on instrument segmentation [13], [15], [28], [6], [29].

B. Objective Functions

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

\[
L_{CE}(s_i, \tilde{s}_i) = -\frac{1}{M} \sum_k (s_i)_k \log(\tilde{s}_i)_k \tag{1}
\]

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 III-C). 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 $\bar{\tilde{s}}_i$. 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

\[
L_{fw} = \frac{1}{N} \sum_{i=1}^{N} L_{CE}(s_i, \bar{\tilde{s}}_i^{i-1}) \tag{2}
\]

\[
L_{bw} = \frac{1}{N} \sum_{i=N}^{1} L_{CE}(s_i, \bar{\tilde{s}}_i^{i+1}) \tag{3}
\]

When the network is trained with real frame sequences, we input the labeled real frame and calculate its cross-entropy loss $L_{1st}$. Assume $x_k$ is the real labeled frame in a sequence, then we have

\[
L_{1st} = L_{CE}(s_k, \tilde{s}_k) \tag{4}
\]

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

\[
L = \lambda_1 L_{fw} + \lambda_2 L_{bw} + \lambda_3 L_{1st} \tag{5}
\]

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

\[
L = \lambda_4 L_{last} + \lambda_5 L_{1st} \tag{6}
\]

where $\lambda$ are hyper-parameters that balance the impact of the losses. We choose $\lambda_1 = \lambda_2 = \lambda_3 = \frac{1}{3}$ and $\lambda_4 = \lambda_5 = \frac{1}{2}$.

### C. Multi-frame Feature Aggregation (MFFA)

MFFA consists of a Temporal Aggregation Block (TAB) and a Spatial Aggregation Block (SAB) in series as shown in Fig. 2. MFFA requires $\tilde{y}_{i-1}$, $h_{i-1}$ and $f_i$ as inputs and outputs $h_i$. In endoscopic sinus surgery, distinguishing instruments from the background can be very challenging due to reflection or the presence of blood as shown in Fig. 3. Because instrument locations are usually close in neighboring video frames, we propose to emphasize the instrument regions of $f_i$ by using the instrument features of $h_{i-1}$ for aggregation in TAB. 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 convolution is followed by a sigmoid function. It estimates the similarity between the previous and current feature maps, and gives 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 $f_t$ to generate $f'_t$ 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 $f_t$ to SAB.

After temporal feature aggregation in TAB, the generated features $f'_t$ is further aggregated using SAB based on spatial relationships. SAB is inspired by the spatial self-attention module [30] and Graph Attention Network (GAT) [31]. Given $f'_t \in \mathbb{R}^{H \times W \times C}$, $f'_t$ can be considered as a graph with $H \times W$ vertices. To estimate the edge values of the graph, we use part of the spatial self-attention module to calculate a spatial attention matrix $A \in \mathbb{R}^{(H \times W) \times (H \times W)}$ [30]. Specifically, $f'_t$ is passed to two parallel 1$\times$1 convolutions for generating two new feature maps $\{f'_{t,b}, f'_{t,c}\} \subset \mathbb{R}^{H \times W \times \frac{C}{2}}$, which are then reshaped to $\mathbb{R}^{(H \times W) \times \frac{C}{2}}$. Next we transpose $f'_{t,b}$ to $\mathbb{R}^{\frac{C}{2} \times (H \times W)}$ and perform matrix multiplication between $f'_{t,b}$ and $A$ to get the spatial attention matrix $A$. Each element of $A$ represents the similarity between the corresponding positions in $f'_t$. Then we reshape $f'_t$ to $\mathbb{R}^{(H \times W) \times C}$ 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 $h'_t$. The expression of this process is given by

$$h'_t = Af'_t \cdot w + b$$

After reshaping $h'_t$ from $\mathbb{R}^{(H \times W) \times C}$ to $\mathbb{R}^{H \times W \times C}$, we further refine the features using a ResNet module-like block proposed in [32]. Specifically, $h'_t$ goes through a 1$\times$1 convolution and the softmax function to generate a coarse segmentation $\tilde{s}_{crs}$. Finally, the output of SAB is obtained by

$$h_t = h'_t + \Phi(h'_t, \tilde{s}_{crs})$$

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

D. 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 [33]. In [33], 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 [34].

Different from [33], we need frame sequences with each frame labeled for training. Given a real frame $x$ and its label $y$, our goal is to generate a 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 in the $i$th frame, we define its translations on the $x$ and $y$-axis and the rotation with respect to the instrument in $x$ as $dx_i$, $dy_i$, and $d\theta_i$, which are called moving parameters.

Then we randomly select the moving parameters for the first and the last frames (i.e., $dx_0, dy_0, db_0, dx_N, dy_N$) and 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 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. Fig. 4 shows examples of synthetic frame sequences.

IV. DATASETS

We evaluate our model on two public datasets of different surgery types: endoscopic sinus surgery dataset UW-Sinus-Surgery-C/L [4] and laparoscopic proctocolectomy dataset RObUST-MIS-Proctocolectomy [13], [35].

**UW-Sinus-Surgery-C/L** consists of a cadaver and a live endoscopic sinus surgery dataset UW-Sinus-Surgery-C/L [4]. The cadaver dataset consists of 10, 5-23 minute cadaveric surgery videos with a resolution of 320$\times$240. A total of 4345 frames were extracted with a sampling rate of 0.5 Hz from the videos. The instrument regions were manually labeled for training semantic segmentation. The live dataset consists of 3, 12-66 minute live surgery videos with a resolution of 1920$\times$1080. Similarly, a total of 4658 frames were extracted with a sampling rate of 1 Hz and labeled.

The main difficulties of this dataset include background reflection on the instrument surface, specular reflection, blur from motion, blood and smoke. 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 instruments are usually used at low speeds during surgery, but senior surgeons who are more familiar with the anatomic structures may make instruments faster, especially when switching surgery sites. Also, smoke generated when electrocauterities are used can make the instruments hard to identify. Considering that these challenging conditions usually are not consistent through the entire video but switch between each other, temporal information could be useful to improve segmentation performance.

**Endoscopic frames** have black border regions that contain no useful information, so we extracted frame center regions and downscaled them to 240$\times$240 for segmentation. To evaluate the generalization ability of our model, we performed K-fold cross-validation with $K = 3$. We use the same fold
split rules as stated in [4]. Specifically, the 10 videos in the cadaver dataset are split into the following 3 folds: 4 videos (Procedure ID: 1, 2, 3, and 4), 3 videos (ID: 5, 6, and 7), and 3 videos (ID: 8, 9, and 10). For the live dataset, the 3 folds are formed by the 1st, 2nd, and the 3rd videos.

ROBUST-MIS was published for MICCAI 2019 Robust Medical Instrument Segmentation challenge, and it consists of a proctocolectomy, a rectal resection, and a sigmoid resection dataset [13, 25]. As an initial exploration, we evaluated our method only on the proctocolectomy dataset for the binary segmentation task. The proctocolectomy dataset has a training set of 2943 960 × 540 labeled frames sampled from 8, 3-5 hour laparoscopic videos corresponding to 8 procedures. The difficulties of this dataset include reflection, blood, smoke, various lighting conditions, small and transparent instruments, and blur from motion. Similarly, we performed 3-fold cross-validation to evaluate our model. Based on the number of frames from each procedure, we split the given training data into the following 3 folds: 6 videos (ID: 1, 2, 3, 4, 5, and 8), 1 video (ID: 9), and 1 video (ID: 10).

V. IMPLEMENTATION DETAILS

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

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. MFFA was implemented based on Fig. 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 [37] 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 on the learning rate 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 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 only the real frame sequences or ii) trained with only the synthetic frame sequences in the first 20 epochs and trained with only the real frame sequences in the last 20 epochs. Details of these two training settings can be found in Section III-B. The second setting is represented with ‘Synth.’ in Table I and Table III. For ROBUST-MIS-proctocolectomy, we only used the real frame sequences for training.

Synthetic and real frame sequence. Considering the computation ability, we train models with sequences of 4 frames. To generate synthetic sequences, we need to decide the translations and rotation angles of the instruments in the first and the last frames 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 y-axis directions. The rotation angles were randomly selected from a uniform distribution over -30 to 30 degrees. Moreover, real frame sequences were extracted around labeled frames in the videos and each frame in a sequence was sampled out of 3 consecutive frames.

VI. EXPERIMENTS AND RESULTS

We combined MFFA with DeepLabV3+ (DL3+) [25] (abbreviated as MFFA-DL3+) and compared with the original version of DeepLabV3+ [25] as baseline. MFFA-DL3+ were also compared with advanced segmentation models include TernausNet [36], LWANet [15] and MAFA [28].

A. Evaluation

For evaluating the segmentation performance, we used Dice similarity coefficient (DSC) and Intersection over Union (IoU), which are defined as

\[
DSC = \frac{2|X \cap Y|}{|X| + |Y|}, \quad IoU = \frac{|X \cap Y|}{|X \cup Y|}
\]

where $X$ and $Y$ are the predicted and ground truth segmentation maps, respectively. Further, we estimate 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 IV 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.

B. Experiments on UW-Sinus-Surgery C/L

The segmentation results on UW-Sinus-Surgery C/L is shown in Table I. 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 differences between the eight models in Group 2 and 3, we performed Friedman test [39] followed by a Conover post hoc test [40] on the
used in experiment 2, while DL3+ with a trimmed backbone MobileNet-p8 was used in experiment 1, while DL3+(MobileNet) was used in experiment C/L with DeepLabV3+ [25], as shown in Table III. The original version of DL3+(MobileNet) was used in experiment 1, while DL3+ with a trimmed backbone MobileNet-p8 was used in experiment 2~8. Because TAB is responsible for length and segmentation performance.

Further, we assessed the sensitivity of segmentation performance to the frame sequence length on UW-Sinus-Surgery-C/L with MFFA-DL3+(MobileNet-p8). Fig. 5 shows the test mDSC with frame sequence length ranges from 2 to 8. We found 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.

C. Experiments on ROBUST-MIS-Proctocolectomy

We compared the performance of DL3+(MobileNet), DL3+(MobileNet-p8), and MFFA-DL3+(MobileNet-p8) on ROBUST-MIS-Proctocolectomy, as shown in Table IV. The models were trained only with real sequences. Compared with DL3+(MobileNet), MFFA-DL3+(MobileNet-p8) achieved better performance on both the cadaver and live datasets, while MFFA-DL3+(MobileNet-p8) on UW-Sinus-Surgery-C/L achieved comparable results to MFFA-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+ [25], as shown in Table III. The yellow line (‘Synth. & Real’) shows the performance of models trained with both synthetic and real sequences.

### Table I: Segmentation Performance on UW-Sinus-Surgery-C/L

| Group | Model(Backbone) | Proposed | Performance (mDSC(%) / mIoU(%) | Time (ms) |
|-------|----------------|----------|-------------------------------|-----------|
|       |                | MFFA    | Sinus-Surgery-C | Sinus-Surgery-L |
| 1     | TernausNet-16 [25] (VGG16) |          | 85.4/2.2 / 89.0/2.7 | 12.9 |
|       | LWANet [19] (MobileNet) |          | 81.1/3.8 / 74.9/4.5 | 5.3 |
|       | MFFA-DL3+ [25] (ResNet50) |          | 91.2/1.0 / 86.8/1.2 | 19.3 |
| 2     | DL3+ (MobileNet) |          | 81.4/4.2 / 75.5/4.8 | 2.1 |
|       | DL3+(MobileNet-p8) |          | 80.4/3.5 / 74.0/4.4 | 2.1 |
|       | MFFA-DL3+(MobileNet-p8) |          | 84.8/2.9 / 79.4/3.3 | 2.9 |
|       | MFFA-DL3+(MobileNet-p8) |          | 86.4/2.2 / 81.1/2.7 | 2.9 |
| 3     | DL3+ (ResNet50-b3) |          | 86.0/2.0 / 81.0/2.5 | 8.2 |
|       | DL3+(ResNet50-b3) |          | 83.9/3.9 / 78.8/5.6 | 5.0 |
|       | MFFA-DL3+(MobileNet-p8) |          | 86.2/2.5 / 81.0/3.0 | 5.4 |
|       | MFFA-DL3+(MobileNet-p8) |          | 85.9/1.4 / 84.0/1.8 | 5.5 |

* i) The mDice and mIoU are shown in ‘mean(standard deviation)’; ii) MobileNet-p8 is the trimmed MobileNet and ResNet50-b3 is the trimmed ResNet50; iii) ‘Synth.’ indicates that the synthetic data are used in the first 20 training epochs; v) The bold font indicates the best performance in the column of each group.

### Table II: Conover Post Hoc Test Results of Baselines (DeepLabV3+ [25]) and Proposed Models on UW-Sinus-Surgery-C/L Dataset

| Compared Model (Backbone) | p-value | mDSC (%) | mIoU (%) | Time (ms) |
|---------------------------|---------|----------|----------|-----------|
|                           |         |          |          |           |
| DL3+(MobileNet-p8,R) |          | 0.83     |          |           |
| MFFA-DL3+(MobileNet-p8,R) |          | 0.08     |          |           |
| MFFA-DL3+(MobileNet-p8,R) |          | 0.02*    |          |           |
| DL3+(ResNet50-b3,R) |          | 0.38     |          |           |
| MFFA-DL3+(ResNet50-b3,R) |          | 0.74     |          |           |
| MFFA-DL3+(MobileNet-p8,R) |          | 0.07     |          |           |
| MFFA-DL3+(MobileNet-p8,R) |          | 0.83     |          |           |
| MFFA-DL3+(MobileNet-p8) |          | 0.51     |          |           |
| MFFA-DL3+(MobileNet-p8) |          | 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) p-values < 0.05 are marked with ‘*’.

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 < 0.05 and the pairwise comparisons calculated using Conover’s test are shown in Table I. In each row of Table I, a p-value < 0.05 indicates that there is a statistically significant difference between the corresponding paired comparison methods.

When both synthetic and real frame sequences are used for training, MFFA-DL3+ with reduced backbone achieved superior performance of 3.1%~9.5% better mDSC and 3.3%~10.7% better mIoU with less inference time compared with the baseline DL3+ [25]. The third row of Table I 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.

![Fig. 5. 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.](image-url)
Fig. 6. Examples of segmentation results. In (a-d), the first frame is the input raw frame and the next four frames are the segmentation results of DeepLabV3+ (DL3+) 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. The example results of UW-Sinus-Surgery-C, UW-Sinus-Surgery-L, ROBUST-MIS-Proctocolectomy are shown in (a,b), (c,d), and (e,f), respectively.

Table III

| No. | Method | Performance (mDSC(%) / mIoU(%) | 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(3.9)/74.4(4.4) | 75.7(7.0)/68.0(8.3)   |
| 3   | MobileNet | 82.4(1.7)/76.6(2.2) | 73.1(5.5)/71.7(7.6)   |
| 4   | MobileNet-p8 | 83.0(3.3)/76.6(4.0) | 77.4(7.0)/69.9(8.4)   |
| 5   | MobileNet | 83.8(1.6)/78.2(2.0) | 80.1(5.5)/79.2(6.6)   |
| 6   | MobileNet-p8 | 84.8(2.9)/79.4(3.3) | 81.3(6.0)/74.7(7.1)   |
| 7   | MobileNet | 84.3(1.7)/78.9(2.3) | 79.8(5.3)/73.3(6.2)   |
| 8   | MobileNet-p8 | 86.4(2.2)/81.1(2.7) | 83.2(4.1)/77.0(5.0)   |

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

Table IV

| Model/Backbone | mDSC (%) | mIoU (%) | Time (ms) |
|---------------|----------|----------|-----------|
| TernausNet-16 | 79.6(2.9) | 71.6(3.0) | 21.6      |
| LWNet | 76.2(3.2) | 67.6(3.3) | 14.2      |
| MAFADL3+ | 81.7(2.8) | 74.1(3.0) | 27.8      |
| DL3+ | 78.1(3.1) | 69.9(3.2) | 9.0       |
| DL3+ (MobileNet-p8) | 78.4(3.8) | 70.4(4.0) | 6.4       |
| MFFA-DL3+ (MobileNet-p8) | 81.0(3.1) | 73.5(3.2) | 8.5       |

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

* The bold font indicates the best performance in the column of the second group (last three rows).

achieved superior performance of 2.5%~3.0% better mDICE and 3.3%~3.8% better mIoU with less inference time.

VII. DISCUSSION

We propose MFFA that aggregates features both temporally and spatially for better segmentation. By distributing the feature extraction burden to each frame in the sequence, our method allows using lightweight encoders to reduce computation costs.

Fig. 6 shows some segmentation results. As shown in Fig. 6(a-d), MFFA could help improve segmentation by reducing false negatives under many challenging conditions including reflection and blood compared to the baseline experiments. Moreover, we found that MFFA could help reduce the false positives on objects that are not parts of the human body such as gauze, as shown in Fig. 6(e). In this work, we focus on semantic segmentation in which pixels are classified into different classes without separating different objects. When analyzing surgical videos that have several instruments 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 for instance segmentation as future work.

Table II shows that the synthetic sequences helped achieve better segmentation performance than when only the real frame sequences were used for training, but Table III shows that this superiority is still not significant enough with p-values of 0.51 and 0.13. There is still space left for improvement because the current synthesis method is relatively simple. We chose this method because i) the sequences consist of only 4 frames with small instrument movements, so this method already provides acceptable approximations to the real sequences. ii) The statistical or other knowledge of instrument moving behaviors are not available in the studied datasets. The true instrument movement pattern could be considered in the future to generate more realistic sequences. One potential option is performing interpolation between two neighboring labeled
frames to generate better synthetic trajectories. On the other hand, the generative adversarial networks should be explored for generating more realistic frames in the synthetic sequences.

VIII. CONCLUSIONS

In this work, we develop and validate a MFFA module that performs feature aggregation based on temporal and spatial relationships between frame pixels to improve instrument segmentation. By using MFFA, we can reduce the deep encoder to its trimmed version and decrease the computation costs. Another advantage of MFFA is that it can be easily combined with any segmentation model that has an encoder-decoder architecture. Also, we propose a simple but effective method that generates synthetic frame sequences to assist network training and compensate for the lack of densely labeled real frame sequences. In the future, we will further improve the sequence synthesis approach and apply the proposed method to surgical video instance segmentation.

REFERENCES

[1] D. Bouget, M. Allan, D. Stoyanov, et al., “Vision-based and marker-less surgical tool detection and tracking: a review of the literature,” Med. Image Anal., vol. 35, pp. 633–654, 2017.
[2] M. Allan, S. Kondo, S. Bodenstedt, et al., “2018 robotic scene segmentation challenge,” arXiv preprint arXiv:2001.11190, 2020.
[3] A. A. Shvets, A. Rahklin, A. A. Kalinin, et al., “Automatic instrument segmentation in robot-assisted surgery using deep learning,” in Proc. IEEE Int. Conf. Mach. Learn. Appl., pp. 624–628, 2018.
[4] S. Lin, F. Qin, Y. Li, et al., “LC-GAN: Image-to-image translation based on generative adversarial network for endoscopic images,” in Proc. IEEE Int. Conf. Intell. Robot. Syst., pp. 2914–2920, 2020.
[5] L. C. García-Peraza-Herrera, W. Li, L. Fidon, et al., “ToolNet: holistically-nested real-time segmentation of robotic surgical tools,” in Proc. IEEE Int. Conf. Intell. Robot. Syst., pp. 5717–5722, 2017.
[6] M. Allan, A. Shvets, T. Kurmann, et al., “2017 robotic instrument segmentation challenge,” arXiv preprint arXiv:1902.06426, 2019.
[7] M. Islam, Y. Li, and H. Ren, “Learning where to look while tracking instruments in robot-assisted surgery,” in Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv., pp. 412–420, 2019.
[8] Q. Chen, X. Zhu, Z.-H. Ling, et al., “Enhanced LSTM for natural language inference,” in Proc. Annu. Meeting Assoc. Comput. Linguistics, pp. 1657–1668, 2017.
[9] G. Liu and J. Guo, “Bidirectional LSTM with attention mechanism and convolutional layer for text classification,” Neurocomputing, vol. 337, pp. 325–338, 2019.
[10] Z. C. Lipton, J. Berkowitz, and C. Elkan, “A critical review of recurrent neural networks for sequence learning,” arXiv preprint arXiv:1506.00019, 2015.
[11] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting,” in Proc. Int. Joint Conf. Artif. Intell., pp. 3634–3640, 2018.
[12] P. Hu, F. Caba, O. Wang, et al., “Temporally distributed networks for fast video semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 8818–8827, 2020.
[13] T. Roii, A. Reinke, P. M. Full, et al., “Comparative validation of multi-instance instrument segmentation in endoscopy: results of the ROBUST-MIS 2019 challenge,” Med. Image Anal., vol. 70, p. 101920, 2021.
[14] M. Attia, M. Hossny, S. Nahavandi, et al., “Surgical tool segmentation using a hybrid deep CNN-RNN auto encoder-decoder,” in IEEE Int. Conf. Syst., Man, Cybern., pp. 3373–3378, 2017.
[15] Z. L. Ni, G. B. Bian, Z. G. Hou, et al., “Attention-guided lightweight network for real-time segmentation of robotic surgical instruments,” in Proc. Int. Conf. Robot. Autom., pp. 9939–9945, 2020.
[16] Y. Jin, K. Cheng, Q. Dou, et al., “Incorporating temporal prior from motion flow for instrument segmentation in minimally invasive surgery video,” in Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv., pp. 440–448, 2019.