JMI Journal of Medical Imaging

RESEARCH PAPER

Inpainting surgical occlusion from laparoscopic video sequences for robot-assisted interventions

S. M. Kamrul Hasan,[a,b,*] Richard A. Simon,[a,c] and Cristian A. Linte[a,b,c]

[a]Rochester Institute of Technology, Biomedical Modeling, Visualization, and Image-guided Navigation (BiMVisIGN) Lab, Rochester, New York, United States

[b]Rochester Institute of Technology, Center for Imaging Science, Rochester, New York, United States

[c]Rochester Institute of Technology, Biomedical Engineering, Rochester, New York, United States

ABSTRACT. **Purpose:** Medical technology for minimally invasive surgery has undergone a paradigm shift with the introduction of robot-assisted surgery. However, it is very difficult to track the position of the surgical tools in a surgical scene, so it is crucial to accurately detect and identify surgical tools. This task can be aided by deep learning-based semantic segmentation of surgical video frames. Furthermore, due to the limited working and viewing areas of these surgical instruments, there is a higher chance of complications from tissue injuries (e.g., tissue scars and tears).

Approach: With the aid of digital inpainting algorithms, we present an application that uses image segmentation to remove surgical instruments from laparoscopic/endoscopic video. We employ a modified U-Net architecture (U-NetPlus) to segment the surgical instruments. It consists of a redesigned decoder and a pre-trained VGG11 or VGG16 encoder. The decoder was modified by substituting an upsampling operation based on nearest-neighbor interpolation for the transposed convolution operation. Furthermore, these interpolation weights do not need to be learned to perform upsampling, which eliminates the artifacts generated by the transposed convolution. In addition, we use a very fast and adaptable data augmentation technique to further enhance performance. The instrument segmentation mask is filled in (i.e., inpainted) by the tool removal algorithms using the previously acquired tool segmentation masks and either previous instrument-containing frames or instrument-free reference frames.

Results: We have shown the effectiveness of the proposed surgical tool segmentation/removal algorithms on a robotic instrument dataset from the MICCAI 2015 and 2017 EndoVis Challenge. We report a 90.20% DICE for binary segmentation, a 76.26% DICE for instrument part segmentation, and a 46.07% DICE for instrument type (i.e., all instruments) segmentation on the MICCAI 2017 challenge dataset using our U-NetPlus architecture, outperforming the results of earlier techniques used and tested on these data. In addition, we demonstrated the successful execution of the tool removal algorithm from surgical tool-free videos that contained moving surgical tools that were generated artificially.

Conclusions: Our application successfully separates and eliminates the surgical tool to reveal a view of the background tissue that was otherwise hidden by the tool, producing results that are visually similar to the actual data.

Keywords: inpainting; segmentation; deep learning; computer vision; poisson blending; optical-flow

Paper 23078GR received Apr. 3, 2023; revised Jul. 21, 2023; accepted Jul. 28, 2023; published Aug. 29, 2023.

*Address all correspondence to S. M. Kamrul Hasan, sh3190@rit.edu

2329-4302/2023/$28.00 © 2023 SPIE

1 Introduction

By drastically lowering the risk of infections and cutting down on hospital stays while still producing results comparable to open surgery, minimally invasive surgery has addressed many of the problems with conventional surgical approaches. Robot assistance in the context of laparoscopic visualization has caused a paradigm shift in this area.[1] The ability to identify surgical instruments is essential for making it easier to manipulate laparoscopic surgical tools while viewing the endoscopic scene. This task of Da Vinci surgical instrument endoscopic segmentation becomes challenging due to the presence of non-class objects, such as suturing threads, as well as other environmental factors such as changing lighting conditions and visual occlusions. These additional elements introduce complexities and make the segmentation task more difficult. The presence of non-class objects requires distinguishing them from the surgical instruments of interest while factors such as lighting variations and occlusions further hinder accurate segmentation. Addressing these challenges requires robust algorithms and techniques that can handle the variability in the visual environment and effectively differentiate between the target surgical instruments and other objects or distractions in the scene.

Endoscopic video is a key visualization approach when performing minimally invasive surgery[2] on a variety of organs.[3] Nevertheless, the endoscopic field-of-view is usually limited, due to the small size of the endoscopic camera, as well as the constrained workspace. This visualization challenge is further augmented by the insertion of surgical instruments into the already limited field-of-view, where surgical tools often occupy a large part of the endoscopic image, even when located near at the edge of the visual field.[4] As a result, surgeons often need to repetitively retract the instruments to observe the tissue, causing significant inconvenience and increasing procedure duration.[5]

Due to the restricted working area and visual field of view, surgical instruments used in endoscopic surgical suturing procedures make it difficult for surgeons to control their dexterity. The likelihood of tissue tears and scars is increased by these obstructions in the visual field. To ensure accurate tracking of the surgical tools and to enable therapy through precise manipulation of the laparoscopic instruments, it is crucial to develop segmentation techniques that are sufficiently accurate and robust. In addition, the issue of tissue occlusion would be mitigated by the transparent rendering or digital removal of the surgical instruments with the appropriate inpainting of the corresponding background information.

While extensive research has focused on the design of smaller, more flexible and higher image quality endoscopic systems,[2,6] very few studies tackled the removal of occlusions caused by the presence of surgical instruments from endoscopic images.[5] Given the extensive reliance on endoscopy to guide access to and provide visualization deep inside the human body, in increasingly narrower spaces, the visual occlusion caused by the surgical instruments poses a severe challenge in need of an effective solution.

In this study, we describe two inpainting approaches for removing the occlusion caused by the presence of surgical instruments in endoscopic images, as well as an automated method for surgical instrument detection, classification, and segmentation, to be used in conjunction with the inpainting algorithms.

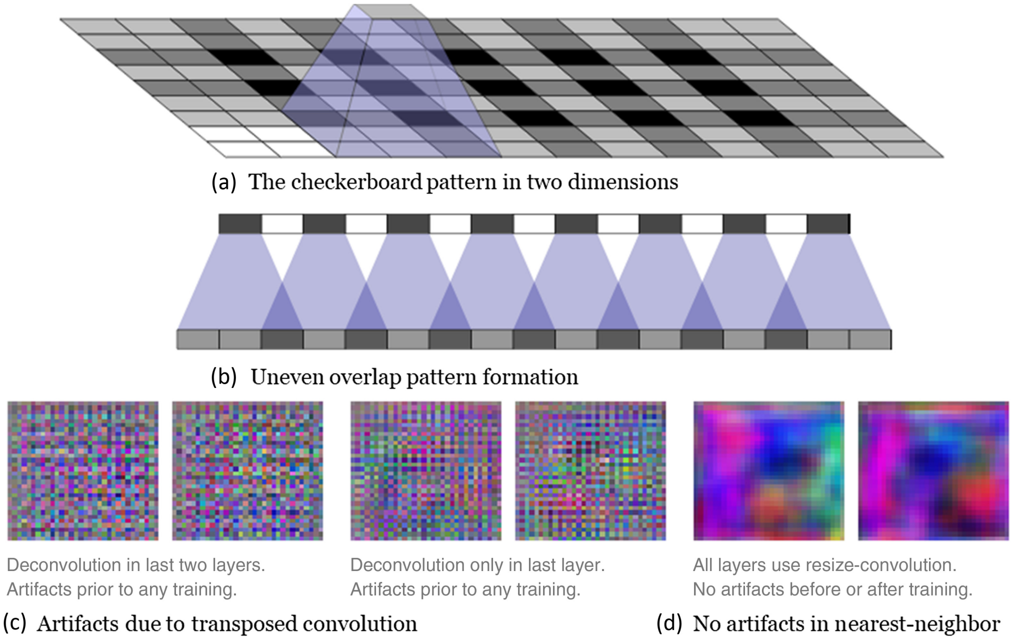
Despite the fact that deep convolutional neural networks (CNNs) have enabled semantic segmentation methods applied to cityscapes, street scenes, and even Landsat image datasets[7,8] in recent years to achieve ground-breaking performance, image segmentation in clinical settings still requires additional accuracy and precision, with even minor segmentation errors posing high risks for effective diagnosis and therapy.

Long et al.[9] proposed the first fully convolutional network for semantic segmentation. The training dataset's small size has made it difficult to use in the medical field, though. To address the aforementioned issue, a number of techniques have been developed, including patch-based training,[10] data augmentation, and transfer learning.[11] However, because there are so many objects in the surgical scene that belong to the same class, semantic segmentation is not accurate enough to handle multi-class objects. As a result, the proposed work is driven by the need to enhance multi-class object segmentation using the strength of the current U-NetPlus and adding new features to it.

U-Net architectures have been widely adopted in the field of medical imaging since 2015, and they have consistently delivered state-of-the-art results for various medical imaging tasks.[12] Recently, Chen et al.[13] modified the U-NetPlus by adding sub-pixel layers to enhance low-light imaging, and they saw promising results with high signal-to-noise ratios (SNRs) and flawless color transformation on their own see-in-the-dark dataset containing 5094 raw short-exposure images, each with a reference long-exposure image.

Jiang and Wang[14] and Jia et al.[15] used nearest-neighbor (NN) interpolation for image reconstruction and super-resolution. The issue of transposed convolution was examined by Odena et al. in their work,[16] which offered a solution using NN interpolation. The value of incorporating it into the deep CNN as a component of the image upsampling operation has not yet been fully examined. A few papers have addressed the challenge of segmenting and identifying surgical instruments from endoscopic video images, and even fewer than half a dozen papers have used deep learning to address this problem. One significant contribution to research has been the use of a modified version of fully convolutional network (FCN)-8, though there have been no attempts at multi-class segmentation.[17] The learned convolution kernel may not be low-pass if the up-sampler is the transposed convolution. In many images, the checkerboard artifact can still be seen. The final image could be overly blurry and thus easily distinguishable from the real images if the lowpass filter removes too much high-frequency content.

Shvets et al.[18] and Pakhomov et al.[19] made the initial proposals for multi-class (instrument part and type) tool segmentation, and both groups reported encouraging outcomes. In a manner similar to the convolutional layers, but in the opposite direction, they modified the traditional U-Net model,[12] which is based on the transposed convolution or deconvolution. As an illustration, they map from 1 input pixel to 4 × 4 output pixels as opposed to mapping from 4 × 4 input pixels to 1 output pixel. The filters' additional weights and parameter requirements necessitate end-to-end training, which significantly slows down computational performance. Transposed convolution can also easily result in "uneven overlap," which is characterized by checkerboard-like patterns and produces artifacts of various scales and colors.[16] The problem with the artifacts and checkerboard patterns produced by the transposed convolution, as shown in Fig. 1, was first described by Redford et al.[20] and Salimans et al.[21] While utilizing a stride of 1 can


(a) The checkerboard pattern in two dimensions

(b) Uneven overlap pattern formation

Deconvolution in last two layers.
Artifacts prior to any training.
Deconvolution only in last layer.
Artifacts prior to any training.
All layers use resize-convolution.
No artifacts before or after training.

(c) Artifacts due to transposed convolution

(d) No artifacts in nearest-neighbor

Fig. 1 Schematic diagram illustrating an artifact caused by the transposed convolution operation: (a) and (b) checkerboard problem caused by applying a transposed convolution on images of improper resolution resulting in uneven overlap, and (c) and (d) artifacts that can be minimized and essentially eliminated by applying a NN interpolation up-sampling operation.

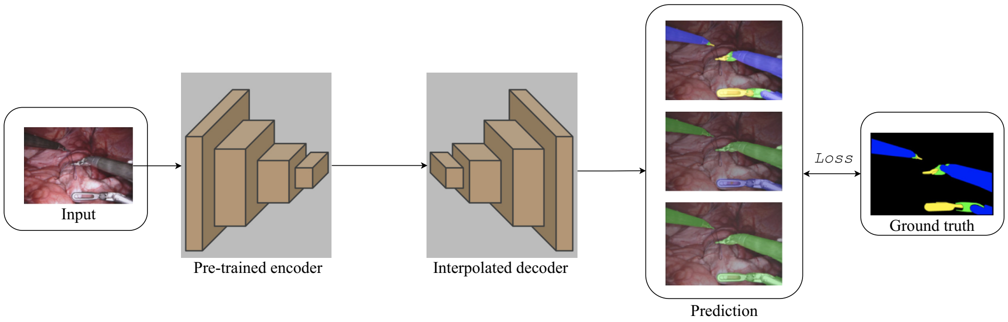


Fig. 2 Pipeline of surgical instruments segmentation.

partially mitigate the issue of uneven overlap in transposed convolutions, it cannot completely eliminate the problem, particularly when using kernel sizes that are divisible by the stride. The reason for the presence of artifacts in the results of the transposed convolutions is their tendency to repeatedly visit pixels in the center of the kernels while only visiting pixels in the corners of their filters once, regardless of how they are configured. This uneven overlap persists even when adjusting filter sizes and stride length. As a result, artifacts are almost inevitable and are likely to be present in the output after several training iterations.

Our aim is to initially reduce the occurrence of these limitations and the resulting artifacts, even though it is challenging to completely eliminate them.

To address these issues with the traditional U-Net architecture, in this work, we introduce the U-NetPlus model in this work by combining the VGG-11 and VGG-16 as encoders, batch-normalizing pre-trained weights, and NN interpolation in place of transposed convolution in the decoder layer (Fig. 2). By avoiding the optimization issues related to the target data, this pre-trained encoder[22] accelerates convergence and produces better results.[23] The artifacts produced by the transposed convolution are also eliminated by the NN interpolation employed in the decoder section.

To evaluate the proposed U-NetPlus network, we implemented several of the latest and most advanced surgical tool segmentation architectures. We then compared the performance of these architectures with that of U-NetPlus. By conducting this comparison, we aimed to assess the effectiveness and superiority of the U-NetPlus architecture in the context of surgical tool segmentation. Only one of the aforementioned papers appears to have obtained results that are comparable to ours,[22] but even this paper still has a number of artifacts, some of which we were able to further reduce using the method we suggested. As a result, even though this paper makes use of some of the fully convolutional network's existing infrastructures, its main goal is to show how existing infrastructures can be modified to improve the performance of the network for a particular task, in this case, the segmentation and identification of surgical instruments from endoscopic images. We show that the decoder can potentially eliminate artifacts and have fewer parameters using NN interpolation.

Furthermore, we demonstrate a novel use of our neural network-based surgical tool segmentor (U-NetPlus) to digitally remove tools from video frames, allowing the visualization of anatomical details that would otherwise be hidden by the tool. According to the authors, there is only one other work that has specifically addressed the segmentation and modification of surgical instruments in endoscopic/laparoscopic videos. This work, presented by Koreeda et al.,[24] proposed a hardware/software-based method to visualize areas that are hidden by surgical instruments. However, their approach has some drawbacks because multiple endoscopes must be present, which might result in more invasive patient care.

In this study, we have implemented and tested two image-driven methods for surgical tool removal. These methods both rely on the use of data from the laparoscope/endoscope images to "paint over" the surgical tool mask that has been detected by our automated surgical tool segmentor. In Fig. 3, we demonstrate two renderings of the background that would normally be hidden behind the surgical tool after the surgical tool has been removed using our suggested application.

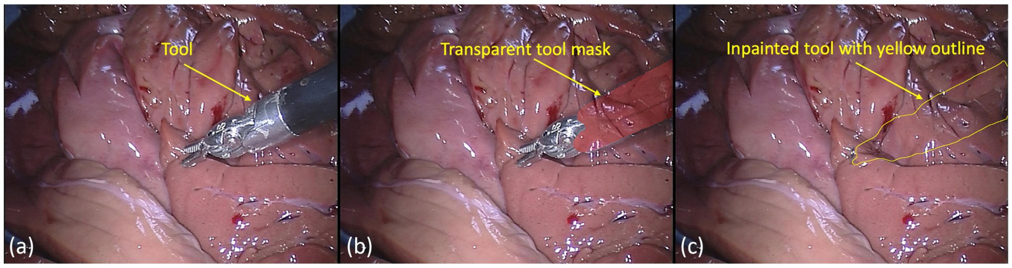


Fig. 3 Example of background renderings generated using our application: (a) tool containing frame; (b) inpainted tool; and (c) inpainted tool with a yellow outline (Video 1, MP4, 192 KB [URL: https://doi.org/10.1117/1.JMI.10.4.045002.s1]).

2 Methodology

2.1 Overview of Proposed Segmentation Method

U-NetPlus has a downsampling path and an upsampling path, followed by a multi-class softmax layer for pixel-wise segmentation, as shown in Fig. 4.

Our prior proposed U-NetPlus[25] functions as an auto-encoder with both a downsampling and an upsampling path, similar to U-Net. Downsampling and upsampling paths are connected through skip connections to keep the number of channels exactly the same as in the encoder portion. This makes it possible for the mask to be very precisely aligned with the original image, which is crucial for medical imaging. The vanishing gradient issue is also mitigated by skip connections by starting multiple paths for backpropagation. To train a network, weights are typically initialized at random. Limited training data, however, can result in overfitting issues, which escalate in cost when the segmentation mask must be manually adjusted. As a result, the network weights can be initialized using transfer learning. Since a surgical instrument is not an ImageNet class, one method of applying transfer learning to a new task is to partially reuse the ImageNet feature extractor (using VGG-11 or VGG-16 as an encoder) and then add a decoder. We started a pre-trained VGG-11 and VGG-16 architecture with batch-normalization layers that have 11 and 16 sequential layers, respectively, as an improvement for the encoder component. After this modification, it has been demonstrated that the pre-trained model can train the network more quickly and accurately.[26]

A rectified linear unit (ReLU) activation function is applied after each of the seven 3×3 kernel-sized convolutional layers that make up the VGG-11 feature map. Utilizing max pooling with stride 1 allowed the feature map to be shrunk in size. After that, the pooling operation

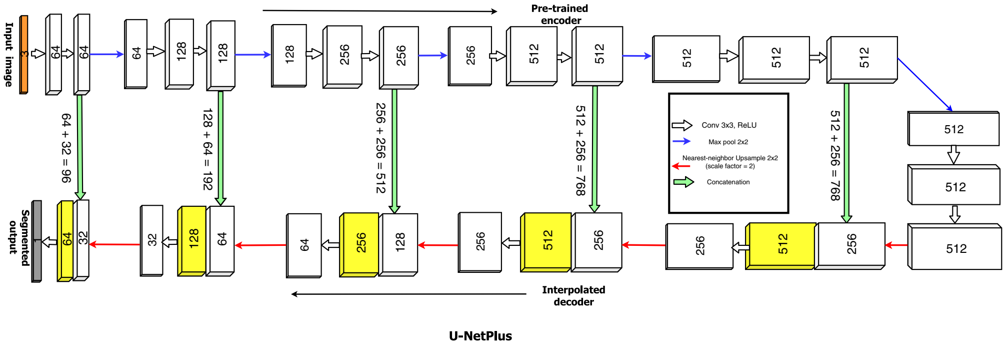
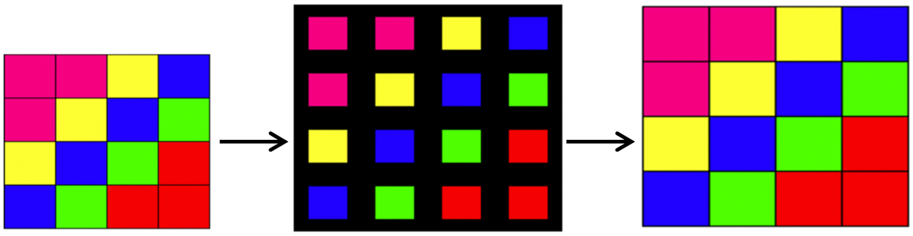


Fig. 4 Modified U-Net with batch-normalized VGG11 as an encoder and upsampling as the decoder. Feature maps are denoted by rectangular-shaped boxes. It consists of both an upsampling and a downsampling path and the feature map resolution is denoted by the box height while the width represents the number of channels. Cyan arrows represent the max-pooling operation, whereas light-green arrows represent skip connections that transfer information from the encoder to the decoder. Red upward arrows represent the decoder, which consists of NN upsampling with a scale factor of 2 followed by 2 convolution layers and a ReLU activation function; working principle of NN interpolation where the low-resolution image is resized back to the original image as shown in Figs. 5(a)–5(c).



(a) **Input size 4 × 4** (b) **Output raster** (c) **Output size 8 × 8**

Fig. 5 (a)–(c) Working principle of NN interpolation where the low-resolution image is resized back to the original image.

increases the number of channels by 2, bringing the total to 512. The pre-trained VGG-11 on ImageNet is used as the source for the weights.

In a recent paper, Santurkar et al.[27] did an investigation into the main impact of batch normalization. This study claims that batch normalization increases training accuracy at a faster convergence rate by re-parameterizing the underlying gradient optimization problem in addition to reducing the internal covariate shift. We applied the BatchNorm layer after each convolutional layer after evaluating the effects of doing so. In contrast to the upsampling path, which results in a pixel-wise mask, the downsampling path reduces the feature size while increasing the number of feature maps, whereas the upsampling path increases the feature size while decreasing the number of feature maps. We altered the current architecture to reconstruct the high-resolution feature maps for the upsampling operation. Instead of transposed convolution, we employed a NN upsampling layer with a carefully chosen stride and kernel size at the start of each block, followed by two convolution layers and a ReLU function that would double the spatial dimension in each block.

The input feature map is upsampled using NN interpolation, which adds a regular grid on top of it. The output grid is created by applying a linear transformation called $\tau_\theta\,(I_i)$ to the input grid Ii, which will be the grid to be sampled. As a result, the definition of τ_θ for an upsampling operation is as follows:

$$\begin{pmatrix} p_i^o \\ q_i^o \end{pmatrix} = \tau_\theta(I_i) = \begin{bmatrix} \theta & 0 \\ 0 & \theta \end{bmatrix} \begin{pmatrix} p_i^t \\ q_i^t \end{pmatrix}, \quad \theta \geq 1, \tag{1}$$

where $(p_i^o, q_i^o) \in I_i$ are the initial sampling input coordinates, (p_i^t, q_i^t) are the target coordinates, and θ is the upsampling factor. Figure 5 illustrates the underlying idea behind NN interpolation, which increases the size of the image. The location of the closest cell center on the input raster is found, and the value of that cell on the output raster is then assigned. This process begins with finding the center pixel of the cell of the output raster dataset on the input raster.

We show the upsampling of a 4×4 image using this method as an illustration. The output raster's cell centers are evenly spaced apart. For each output cell, a value must be extracted from the input raster. The input raster's cells whose centers are closest to the output raster's are chosen for interpolation using the NN method. Copies of the center pixel can be used to fill in the black spaces in the middle image. Therefore, compared to strided or transposed convolution, these fixed interpolation weights require no learning for upsampling operation, resulting in a more memory-efficient upsampling operation. The algorithm is comparable to that suggested and applied by Dong et al.[28] in their work.

2.2 Surgical Tool Removal Method A: Optical Flow-Based Video Object Removal Algorithms

The first method uses video object removal algorithms[29,30] to replace the segmented tool pixels in the current frame with data from previous frames, which we have mentioned in our prior work.[31] The procedure establishes dense correspondences (optical flow) between the pixels (i.e., regions) observed in the background region of a previous frame $I_t(x, y)$ and the pixels (i.e., regions) occluded by the surgical tool in the present frame $I_{t-1}(x, y)$. The foreground surgical tool region Ω_F obscures pixels in the background region Ω_B. The optical flow is used to modify the cumulative mapping function $\mathbf{V}_t(x, y)$, which establishes the correspondences between the foreground pixels in the current frame t and the background pixels in the previous frames $\{I_1, I_2, \ldots, I_{t-1}\}$. After that, the tool region can be painted using this function with data from earlier frames.

A parametric warp model,[32] such as an affine warp, which is defined as the solution to the following minimization problem, can be used to determine the correspondences between the frames

$$\min_{p} \sum_{\substack{x,y\in\Omega_B^t,\\ (x,y)\neq\Omega_F^t,\\ (x+u,y+v)\neq\Omega_F^{t-1}}} [I_t(x,y) - I_{t-1}(x + u(x,y;\mathbf{p}), y + v(x,y;\mathbf{p}))]^2, \tag{2}$$

where

$$\begin{bmatrix} u(x,y;\mathbf{p}) \\ v(x,y;\mathbf{p}) \end{bmatrix} = \begin{bmatrix} p_1 & p_3 & p_5 \\ p_2 & p_4 & p_6 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{3}$$

represents the displacement vector at pixel (x,y) from I_t to I_{t-1} and Ω_B represents the background region used to determine the affine parameters $\mathbf{p}$. The displacement field in the missing tool region Ω_t is determined by evaluating Eq. (3) within the region Ω_t using the determined affine parameters $\mathbf{p}$.

Alternatively, the correspondences can be determined by a non-parametric optical flow-based model[33] as the variational minimization of the following problem

$$\min_{u,v} \sum_{\substack{x,y\in\Omega_B^t,\\ (x,y)\neq\Omega_F^t,\\ (x+u,y+v)\neq\Omega_F^{t-1}}} [I_t(x,y) - I_{t-1}(x + u(x,y), y + v(x,y))]^2 + \alpha(|\nabla u(x,y)|^2 + |\nabla v(x,y)|^2), \tag{4}$$

where α is the weight between the data (first) and smoothness (second) term. The data term represents the similarity between the pixel values of adjacent frames while the smoothness term enforces the smoothness of the flow fields. The data term is undefined inside the tool regions Ω_t^F and Ω_{t-1}^F, so the smoothness term becomes the only constraint resulting in the optical flow field being smoothly interpolated into the missing tool region. We solve both Eqs. (2) and (4) using a multi-resolution (coarse-to-fine) Gaussian pyramid framework.

The most straightforward way to inpaint the tool region of frame Ω_t^F is to use the correspondences (u,v) to trace the backward displacement at each pixel of the tool region Ω_t^F to find its corresponding location in a previous inpainted frame. The occluded pixel in Ω_t^F is then replaced by the corresponding pixel in Ω_{t-1}^F using bilinear interpolation. The current inpainted frame t is then used as a source frame to inpaint the tool region in the next frame Ω_{t+1}^F. A potential problem with this simple inpainting approach can occur when the same anatomical features are covered by the tool for multiple frames. This can result in the inpainted regions becoming blurry due to the repeated copying (via bilinear interpolation) of pixels from the inpainted tool region into the tool region of consecutive frames. This occurs when the tool dwells over or moves slowly across a region covered by the tool.

To avoid this problem, we define a cumulative mapping function $\mathbf{V}_t(\mathbf{x})$,[29,30] which, for each pixel, stores the index of the source frame $I_1, I_2, \ldots, I_{t-1}$ and the spatial shift relative to the source background region where the pixel was last visible. This mitigates the blurriness problem because source pixels used to inpaint the tool region are now being copied once via interpolation as opposed to multiple times. Letting $\mathbf{x} = (x,y)$ and $\mathbf{w} = (u,v)$, the vector field $\mathbf{V}_t$ can be computed for each pixel $\mathbf{x} \in \Omega_t$ using the optical flow $\mathbf{w}$ for frame $t \to t-1$ by propagating the previous frame vector-field value $\mathbf{V}_{t-1}$ $(\mathbf{x} + \mathbf{w}(\mathbf{x}))$ using the following rule

$$\mathbf{V}_t(\mathbf{x}) = \begin{cases} [\mathbf{w}(\mathbf{x}), t-1] & \text{if } \mathbf{x}_{t-1} \notin \Omega_{t-1}^F \\ [\mathbf{w}(\mathbf{x}) + \mathbf{V}_{t-1}^1(\mathbf{x}_{t-1}), \mathbf{V}_{t-1}^2(\mathbf{x}_{t-1})] & \text{if } \mathbf{V}_{t-1}(\mathbf{x}_{t-1}) \neq \text{undefined} \\ \text{undefined} & \text{otherwise} \end{cases}, \tag{5}$$

where $\mathbf{x}_{t-1} = \mathbf{x} + \mathbf{w}(\mathbf{x})$ is the corresponding pixel in the previous frame and $\mathbf{V}^1$ denotes the spatial-shift value (first element) and $\mathbf{V}^2$ denotes the index of the source frame (second element). These rules are applied to pixels covered by the tool in frame t. The first condition occurs if the foreground (occluded) pixel maps back to the background region in frame $t-1$. The second condition occurs if the foreground (occluded) pixel maps back to the foreground region in frame $t-1$ and $\mathbf{V}_{t-1}(\mathbf{x}_{t-1})$ is defined. The last condition indicates that the foreground pixel has not been observed in the background of any previous frames and thus the mapping function is undefined.

2.3 Surgical Tool Removal Method B: Reference Image Frame Inpainting Flow-Based Video Object Removal Algorithms

This method depends on gathering a number of reference image frames prior to the introduction of the surgical instruments into the operating room and into the laparoscope/field endoscope's of view. The inpainting algorithm then uses these reference images $R_i(x, y)$ in place of the segmented surgical tools.

The technique establishes correspondences between regions observed in a reference frame and regions not occluded by the surgical tool Ω_t in the current frame $I_t(x, y)$. We select the closest matching reference frame from the collection of frames recorded prior to the introduction of the tools, further spatially transform it to match the current image, and then fill the tool mask region with the pixels from the warped reference image. To determine the reference image for the current frame, we first use Eq. (3) to identify the reference image with the lowest sum of the square differences between the reference and the current image within a region of interest surrounding the tool mask Ω in the current image using Eq. (6)

$$\min_i \sum_{x\in\Omega^B} [R_i(x, y) - I_t(x, y)]^2, \tag{6}$$

where the reference frame's index i is used. The tool mask's surrounding area and the selected reference must maintain spatial continuity, which is enforced by this term. The chosen reference frame is then transformed spatially to enhance its registration with the current frame and to ascertain the displacement field in the missing tool region. The spatial transformations can be specified by either a non-parametric optical flow-based model Eq. (4) or an affine parametric motion model Eq. (2), which is similar to the previous method A.

2.4 Illumination/Appearance Adjustment

The appearance of the same tissue varies across frames as a result of the operating room's uneven lighting. Because of this, there may be obvious seams between the inpainted and existing regions when pixels from the reference images or previous frames are copied into the tool mask region. We employ a Poisson blending algorithm[34] to combine the inpainted tool region with the current frame background I^B to reduce the appearance of seaming artifacts. The gradient fields of the two regions are combined rather than the pixels from the two regions. The formulation follows a variational issue

$$\min_I \sum_{x,y\in\Omega_t^F} |\nabla I(x, y) - \mathbf{v}(x, y)|^2 \text{ with } I^B|_{\partial\Omega} = I|_{\partial\Omega}, \tag{7}$$

where I is the Poisson blended inpainted tool image, $\mathbf{v}$ is the gradient of the inpainted tool image determined by the tool removal algorithms, $\partial\Omega$ is the boundary between the inpainted region, and the background, and Ω_t^F is the tool mask region. The current image provides Dirichlet boundary conditions $I_B|_{\partial\Omega} = I|_{\partial\Omega}$ for the equation around the inpainted region. The solution of Eq. (5) is given by

$$\Delta I(x, y) = \operatorname{div} \mathbf{v}(x, y) \text{ with } I^B|_{\partial\Omega} = I|_{\partial\Omega}, \tag{8}$$

for all $x, y \in \Omega_t^F$ and outside of Ω_t^F I takes on the same values of I^B. This allows the Poisson inpainted region to have intensities similar to the background's boundary with variations corresponding to the gradient $\mathbf{v}$ of the inpainted tool image.

Several frames will pass after the laparoscopic/endoscopic procedure (Video 2) begins before enough anatomical details are revealed to paint the entire tool region. Depending on how quickly the surgical tool is moving, the number of frames will change. If the inpainted region does not fill the entire tool region, Neumann boundary conditions, $\frac{\partial I}{\partial n} = 0$ apply to the pixels bordering the remaining unfilled tool region, with n being the unit normal to the boundary between the inpainted and unfilled tool regions. This will stop the inpainted region from picking up any of the tool's intensities.

It is possible to produce artifacts due to illumination variation of the data used to inpaint the tool region because the data used to inpaint a given tool region comes from multiple previous frames. Gradients within the tool region may result from variations in illumination rather than anatomical structures. The Poisson blending algorithm will not change these internal gradients.

Using the following heuristic rule, we set the div $\mathbf{v}(x, y) = 0$ at locations where nearby inpainted pixels originated from source frames that are more than 10 frames apart to eliminate gradients brought on by illumination differences within the inpainted region. The modified Poisson blending algorithm is the name we give to this technique for eliminating these internal gradients.

2.5 Image Dataset

We utilized the Robotic Instruments dataset from the sub-challenge of the MICCAI 2017 Endoscopic Vision Challenge[35] for both training and validation. The high-resolution stereo camera images were taken from a da Vinci Xi surgical system during laparoscopic cholecystectomy procedures and were collected as 8×225 frame sequences with a 2 Hz frame rate for the training dataset. To prevent any redundancy problems, the frames were re-sampled from 30 Hz video to 2 Hz. The video sequences, which have a resolution of 1920×1080 in RGB format, were recorded using a stereo camera and include the left and right eye views. A rigid shaft, wrist, and claspers were manually labeled on each frame to identify the surgical instrument. The test set consists of 2×300 frames and 8×75 frame sequences. The segmentation of the seven classes—which include grasping retractors, needle drivers, prograsp forceps, vessel sealers, etc.—represents the main difficulty. We created videos with surgical tools from surgical tool-free videos by inserting a moving surgical tool into the surgical tool-free video. The surgical tool-free videos were taken from the Hamlyn Centre Laparoscopic/Endoscopic Video Dataset. The Hamlyn Dataset[36] comprises rectified stereo images with a resolution of 384×192 pixels. These images were collected during partial nephrectomy procedures and do not include camera calibration information. From this dataset, we collected 2 video clips, each containing 600 pairs of stereo images. The duration of each video clip is ∼10 s. To enhance the visualization and inpainting of the videos, we utilized multiple tools to superimpose additional elements on the video frames.

2.6 Data Augmentation

We added data to the MICCAI 2017 EndoVis Challenge using the argumentation library, which was described as a quick and adaptable implementation for data addition in Ref. 37. These libraries contain affine and elastic transformations as well as their effects on the added image data.

In short, the affine transformation includes scaling, translation, horizontal flip, vertical flip, random brightness, noise addition, etc. For the elastic transformation (non-affine), first, a random displacement field, $F(R)$ is generated for the horizontal and vertical directions, δx, and δy, respectively, where $[\delta x, \delta y] = [-1 \leq \delta x, \delta y \leq +1]$.

These random fields are then convolved with an intermediate value of σ (in pixels), and the fields are multiplied by a scaling factor α that controls the intensity. Thus, we obtain the elastically transformed image in which the global shape of the interest is undisturbed, unlike in the affine-transformed image. In an elastic transformation, the deformation applied to the image is typically local, allowing for localized adjustments while maintaining the global shape intact. This means that the essential arrangement and relationships between objects or structures, such as their relative positions, orientations, or proportions, remain relatively unchanged. We use the data augmentation strategies to generate ∼1000 images for training our methods.

2.7 Implementation Details

We utilized PyTorch (https://github.com/pytorch/pytorch) to put our methodology into practice. We removed the unwanted black border from each video frame during the pre-processing stage. By dividing by the standard deviation and subtracting the mean from the images, they were normalized (i.e., according to their z-scores). Prior to each weighted layer, batch normalization was applied because it re-parameterizes the underlying gradient optimization problem and speeds up training convergence.[27] With a learning rate of 0.00001, we used the Adam optimizer for training. Dropout was not used because in our situation, it reduced the performance of validation. All models underwent 100 training iterations. Before each epoch, the training set was randomly shuffled with a batch size of 4. All tests were performed on a computer with an NVIDIA GTX 1080 Ti GPU (11 GB).

2.8 Evaluation Metrics

2.8.1 *Tool segmentation evaluation metrics*

In this work, we used the common Jaccard index—also referred to as the intersection-over-union (IoU)—to evaluate segmentation results. It is an overlap index that quantifies the agreement between two segmented image regions: a ground truth segmentation and the predicted segmentation method. Given a vector of ground truth labels T_1 and a vector of predicted labels P_1, IoU can be defined as [Eq. (9)]

$$J(T_1, P_1) = \frac{|T_1 \cap P_1|}{|T_1 \cup P_1|} = \frac{|T_1 \cap P_1|}{|T_1| + |P_1| - |T_1 \cap P_1|}, \tag{9}$$

where given a pixel j, the label of the pixel z_j, and the probability of the same pixel for the predicted class $\hat{z}_j$, Eq. (9) for k number of dataset

$$J = \frac{1}{k} \sum_{j=1}^{k} \left(\frac{z_j \hat{z}_j}{z_j + \hat{z}_j - z_j \hat{z}_j} \right). \tag{10}$$

We can represent the loss function in a common ground of log scale as this task is a pixel classification problem. So, for a given pixel j, the common loss can be defined as the function H for k number of dataset

$$H = -\frac{1}{k} \sum_{j=1}^{k} (z_j \log \hat{z}_j + (1 - z_j) \log(1 - \hat{z}_j)). \tag{11}$$

From both Eqs. (10) and (11), we can combine and can get a generalized loss

$$L = H - \log J. \tag{12}$$

Our aim is to minimize the loss function, and, to do so, we can maximize the intersection, J between the predicted mask and the ground truth.

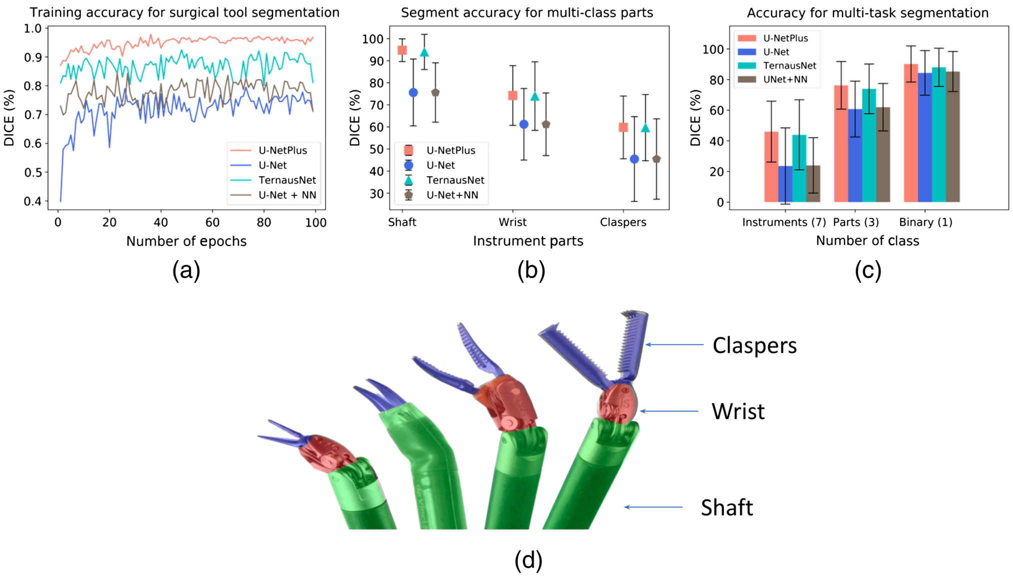
Another commonly used performance metric is the DICE coefficient. Given the set of all pixels in the image, the set of foreground pixels by automated segmentation S_1^a, and the set of pixels for ground truth S_1^g, the DICE score can be compared with $[S_1^a, S_1^g] \subseteq \Omega$, when a vector of ground truth labels T_1 and a vector of predicted labels P_1

$$D(T_1, P_1) = \frac{2|T_1 \cap P_1|}{|T_1| + |P_1|}. \tag{13}$$

DICE score will measure the similarity between two sets, T_1 and P_1, and $|T_1|$ denotes the cardinality of the set T_1 with the range of $D(T_1, P_1) \in 0, 1$.

2.8.2 *Tool inpainting evaluation metrics*

In this work, we report the quantitative evaluation of the inpainted videos using common metrics, including mean squared error (MSE), peak SNR (PSNR), and structural similarity index (SSIM) as image quality metrics. It can be noted that MSE and PSNR are not always well-correlated with perceived/subjective visual quality, whereas SSIM can show better correlations.



(a) (b) (c)

(d)

Fig. 6 Quantitative comparison of (a) training accuracy, (b) multi-class (class = 3) instrument parts, (c) multi-task segmentation accuracy, and (d) different parts (claspers, wrist, and shaft) of the instruments to clarify the comparison.

3 Results

3.1 Quantitative Segmentation Results

We performed a paired comparison between the segmentation results obtained using the traditional U-Net architecture, U-Net + NN, TernausNet, and U-NetPlus to demonstrate the potential improvement in segmentation performance using NN interpolation (i.e., fixed upsampling) in the decoder (our proposed method).

Figure 6 displays training accuracy for binary segmentation over 100 epochs. We contrast our suggested architecture with U-Net, U-Net + NN, and TernausNet, three additional models. The training accuracy of the traditional U-Net framework (shown in blue) with the transposed convolution in the decoder improves as the NN is added to the U-Net decoder. In addition, compared to TernausNet, the training of our suggested method (U-NetPlus) converges more quickly and produces better training accuracy (shown in cyan). Therefore, just this graph shows how the NN interpolation improves segmentation performance.

The MICCAI 2017 EndoVis dataset served as the model's testing ground. The performance of our suggested U-NetPlus framework in comparison to a number of multi-task segmentation techniques is summarized in Table 1. The table shows unequivocally how segmentation across all frameworks—U-Net and TernausNet—improved after NN interpolation was added in the decoder step. In addition, our model was compared to ToolNetH, ToolNetMS, FCN-8s, and concurrent segmentation and localization (CSL), four additional structures aside from U-Net and TernausNet. The last one (CSL) was the pioneering method for segmenting multiple classes of surgical instruments. However, they only used the wrist class that we introduced in our approach and only used the two instrument classes (shaft and claspers). As a result, our overall accuracy was significantly higher than that of the CSL approach.

We used a paired statistical test to assess how well each of these methods (U-Net, U-Net + NN, TernausNet, and U-NetPlus) segmented data using the IoU and DICE metrics. For the purpose of illustration, we conducted a comparison of binary segmentation using different architectures. Our proposed U-NetPlus architecture yielded a statistically significant (for statistical significance testing, Wilcoxon signed-rank test is performed) 11.01% improvement ($p < 0.05$) in IoU and 6.91% DICE ($p < 0.05$) over the classical U-Net framework; a statistically significant 8.0% improvement ($p < 0.05$) in IoU and 5.79% DICE ($p < 0.05$) over the U-Net + NN framework; a statistically significant 0.18% improvement in IoU and 0.21% DICE ($p < 0.1$) over the state-of-the-art TernausNet framework[18] for the binary segmentation.

Table 1 Quantitative comparison for instrument segmentation across several techniques from test set.

	Metric					
	Binary segmentation		Instrument part		Instrument type	
Models	IoU	DICE	IoU	DICE	IoU	DICE
ToolNetH[17]	74.4	82.2	—	—	—	—
ToolNetMS[17]	72.5	80.4	—	—	—	—
FCN-8s[17]	70.9	78.8	—	—		—
CSL[38]	—	88.9	—	87.70 (shaft)	—	—
U-Net[12]	75.44	84.37	48.41	60.75	15.80	23.59
Std. dev.	±18.18	±14.58	±17.59	±18.21	±15.06	±19.87
U-Net + NN	**77.05****	**85.26***	**49.39***	**61.98***	**16.72***	**23.97**
Std. dev.	±15.71	±13.08	±15.18	±15.47	±13.45	±18.08
TernausNet[18]	83.60	90.01	65.50	75.97	33.78	44.95
Std. dev.	±15.83	±12.50	±17.22	±16.21	±19.16	±22.89
U-NetPlus-VGG-11	81.32	88.27	62.51	74.57	**34.84***	**46.07****
Std. Dev.	±16.76	±13.52	±18.87	±16.51	±14.26	±16.16
U-NetPlus-VGG-16	**83.75**	**90.20***	**65.75**	**76.26***	34.19	45.32
Std. dev.	±13.36	±11.77	±14.74	±13.54	±15.06	±17.86
				94.75 (shaft)		

Mean and (standard deviation) values are reported for IoU (%) and DICE coefficient (%) from all networks against our proposed U-NetPlus. The statistical significance of the results for U-Net + NN and U-NetPlus model compared against the baseline model (U-Net and TernasuNet) are represented by * and ** for *p*-values 0.1 and 0.05, respectively. U-Net has been compared with U-Net + NN, TernausNet has been compared with U-NetPlus. The best performance metric (IoU and DICE) in each category (binary, instrument part, and instrument type segmentation) is indicated in bold text.

By assigning the corresponding index from the training set to each instrument pixel, multi-class instrument segmentation was carried out. Shaft, wrist, and claspers were the three classes that made up this application. The multi-class segmentation using our suggested U-NetPlus framework produced a mean IoU and DICE of 65.75% and 76.26%, respectively. Figure 6 shows how the U-NetPlus architecture compares to the other three frameworks in terms of accuracy and precision. As demonstrated, the U-NetPlus framework outperforms TernausNet, the framework that is currently regarded as best in class. The IoU metric follows the same trends as the DICE score, as shown in Table 1.

According to the training set, the instrument type was segmented by assigning the appropriate instrument type to each instrument pixel while assigning the value 0 to all background pixels. U-NetPlus-VGG-11 encoder performed better than U-NetPlus-VGG-16 in the segmentation of instrument type (for class = 7) cases. We can improve upon our instrument-type segmentation results. In instrument type segmentation, VGG-16, being a deeper and more complex architecture with a larger number of parameters, may face challenges when operating on smaller datasets. The limited data could lead to overfitting, affecting its performance. On the other hand, VGG-11, with a shallower and simpler architecture, gains an advantage in such scenarios as it is less prone to overfitting. In addition, instrument type segmentation may not require highly intricate features, leading to VGG-11 generating more straight forward representations better suited for the task.

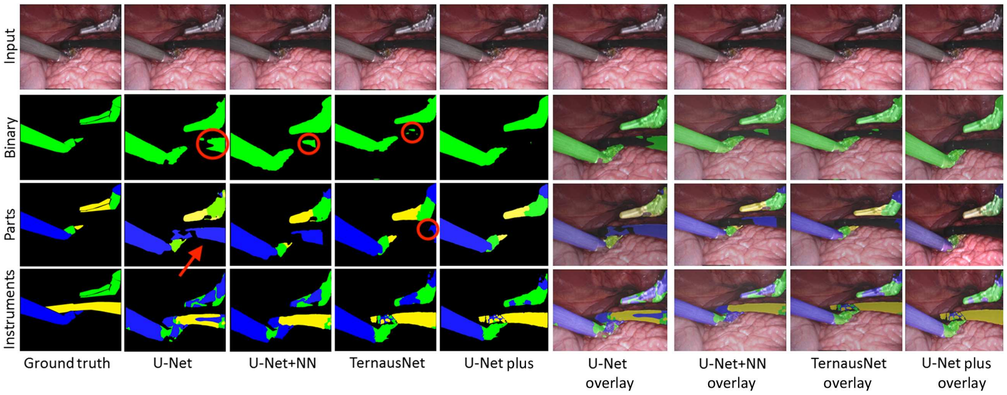


Fig. 7 Qualitative comparison of binary segmentation, instrument part and instrument type segmentation result and their overlay onto the native endoscopic images of the MICCAI 2017 EndoVis video dataset yielded by four different frameworks: U-Net, U-Net + NN, TernausNet, and U-NetPlus.

3.2 Qualitative Segmentation Results

Figure 7 shows the qualitative comparison of our suggested model for both binary and multiclass instrument segmentation. The second row of the figure demonstrates that the classical U-Net for binary segmentation displays an aspect of the instrument that was absent from the binary mask of our ground truth data (second row and second column). When it comes to binary segmentation, U-netPlus performs the best (i.e., it can clearly separate the instruments from the background) while TernausNet still leaves some unwanted regions in the segmentation output.

U-NetPlus can segment the three classes (blue: shaft, green: wrist, and yellow: claspers) nearly perfectly compared to TernausNet when it comes to segmenting instrument parts. U-Net still segments the unwanted instrument (blue) in this case. When it comes to segmenting instruments by type, it is obvious that U-Net cannot distinguish between the blue and green classes, whereas TernausNet and U-NetPlus can do so more successfully. According to Fig. 7, the figure clearly shows that U-NetPlus outperforms U-Net, U-Net + NN, and TernausNet in terms of quality.

3.3 Segmentation Ablation Study

To examine the segmentation performance further, we carried out an additional ablation analysis. This attention study employed a state-of-the-art image saliency technique[39] to determine the regions of interest in an image. The technique utilizes a method of suppressing the softmax probability of the target class to learn a mask for the image. By applying this approach, the study was able to reveal where our proposed algorithm focuses its attention within an image, providing insights into the areas that are deemed significant for the algorithm's decision-making process. The segmented surgical instruments' heat-map image is superimposed onto the original video image in Fig. 8.

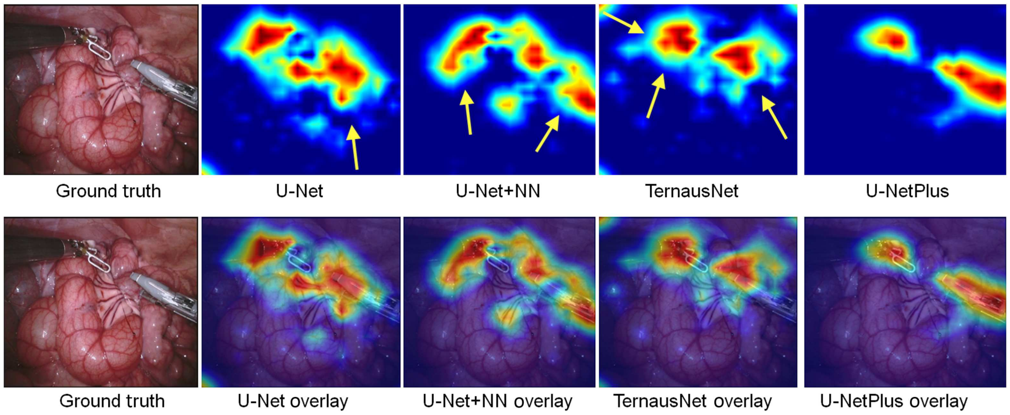


Fig. 8 Attention results: U-NetPlus "looks" at a focused target region, whereas U-Net, U-Net + NN and TernausNet appear less "focused," leading to less accurate segmentation.

The U-Net + NN architecture with NN sampling in the decoder path and the conventional U-Net encoder outperformed the conventional U-Net architecture, as shown in Fig. 8 (featuring the transposed convolution in the decoder). However, the U-Net + NN framework slightly underperformed the TernausNet architecture with the pre-trained VGG network in the encoder because of the small training dataset. In contrast to the conventional U-Net, U-Net + NN, and TernausNet frameworks, our suggested approach (U-NetPlus) localizes the wrist and claspers of the bipolar forceps almost perfectly using this class activation mapping (Fig. 8). As a result, better overall performance is obtained by skillfully integrating and combining NN interpolation as a fixed upsampling technique with a pre-trained encoder.

3.4 Tool Removal Results

3.4.1 *Tool removal results: method A*

The first surgical video serves as an example of how our tool segmentor can effectively segment and produce a mask that can be used to eliminate the tool from the video images. While viewing the anatomy *in vivo* and with little surface distortion in this video, the camera is stationary. The outcomes of the tool segmentor [Figs. 9(a)–9(c), yellow outline] and tool removal method A, which inpaints the segmentation mask region using an affine parametric motion model, are shown in Fig. 9. Most frames display tool segmentation outcomes that are similar to those shown in Figs. 9(a), 9(c), 9(d), and 9(f). On occasion, as in Figs. 9(b) and 9(e), the tool segmentor misses a portion of the tool claspers.

The segmentation mask was dilated by 20 pixels to account for under-segmentation and to ensure complete inpainting of the tool. The frame that occurred early in the process (Video 1) when insufficient anatomical information had been uncovered to completely inpaint the tool region is what led to the incomplete inpainting results in Figs. 9(a) and 9(d).

We created videos with surgical tools from surgical tool-free videos by inserting a moving surgical tool into the surgical tool-free video to test our tool removal algorithms on more challenging cases where the camera and/or anatomy are in motion. The ground truth mask was used during surgery, and the surgical tool was taken from the MICCAI 2015 dataset. The surgical tool-free videos were taken from the Hamlyn Centre Laparoscopic/Endoscopic Video Datasets. In these situations, the ground truth mask was used to create the tool segmentation mask, which was then dilated by 1 pixel.

In Fig. 10, we demonstrate representative examples of tool removal method A using an affine parametric motion model to remove the tool from a moving video of a porcine abdomen with the least amount of abdominal deformation. The tool containing the frame is shown in Fig. 10(a), the modified Poisson blended inpainting results are shown in Fig. 10(b), and the

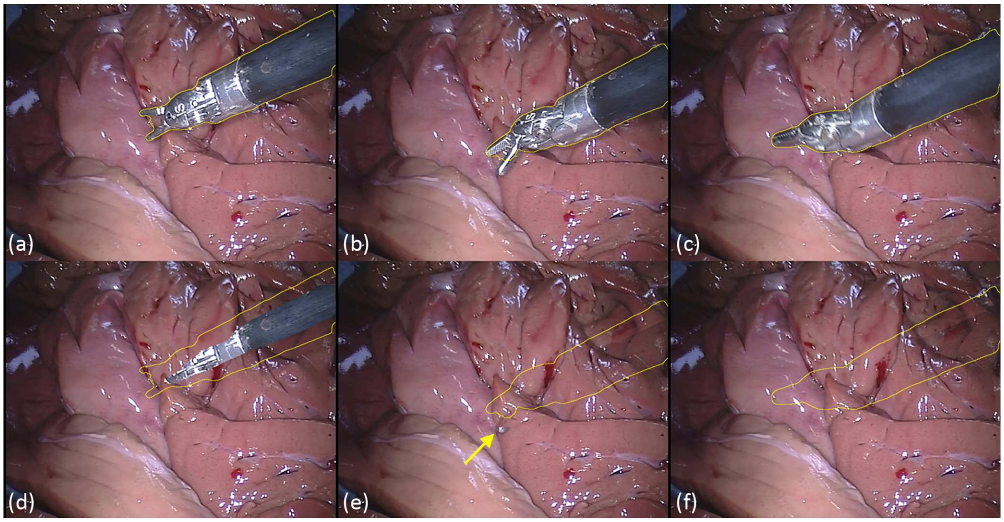


Fig. 9 (a)–(c) Tool containing frames with U-NetPlus segmentation results (yellow outline). (d)–(f) Inpainted results using method A; yellow arrow in mid-column shows residual tool caliper.

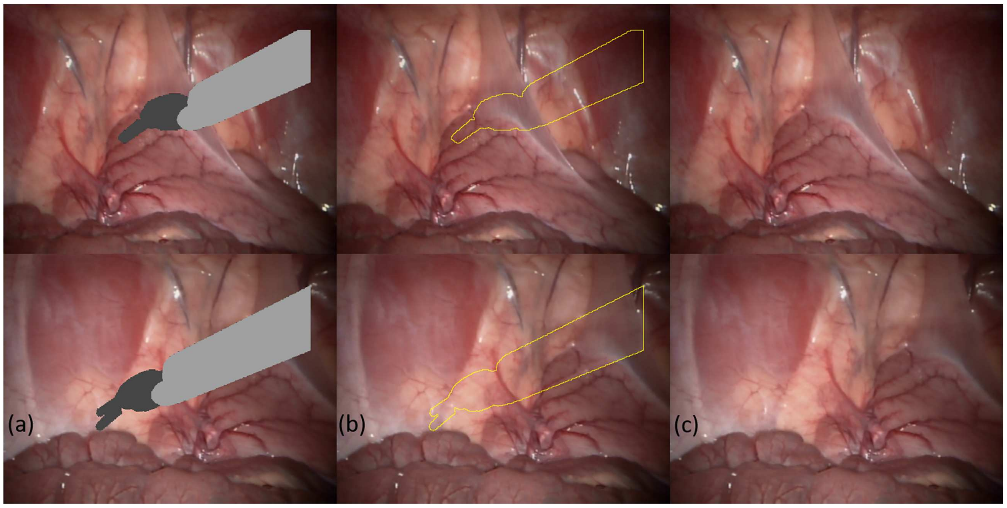


Fig. 10 Two examples showing tool removal method A with an affine parametric motion model: (a) tool containing frames; (b) modified Poisson blended inpainted results; and (c) ground truth frames.

ground truth is shown in Fig. 10(c). It can be seen that using method A along with the modified Poisson blending algorithm yields outcomes that are visually similar to the actual data.

We demonstrate in Fig. 11 the effectiveness of the modified Poisson blending algorithm in reducing internal illumination seams. Because the data used to inpaint a given tool region are compiled from several prior frames, there is a chance that illumination variations will cause artifacts to appear. As a result, gradients within the tool region may appear that are not caused by anatomical structures but rather by variations in illumination in the data used to inpaint the tool region.

Figure 11(a) shows frame containing a tool where the grayscale values inside the tool correspond to the source frames used to inpaint the tool, and Fig. 11(b) shows a plot of the source frame versus distance along the red line in Fig. 11(a). The yellow arrow points to a region where there is a temporal discontinuity between the source frames used to inpaint the tool region. As shown in Fig. 11(c), these internal gradients will persist after applying the Poisson blending algorithm. In Fig. 11(d), we applied the modified Poisson blending algorithm where the internal gradients are suppressed by setting div $\mathbf{v}(x, y) = 0$ in Eq. (8) at locations where neighboring inpainted pixels originated from frames that are >10 frames apart.

In Fig. 12, we contrast applying the cumulative to simple mapping functions to inpaint the tool region where the tool moves slowly across a region where the same anatomy is covered for several frames. The results of the inpainted tool using the straightforward noncumulative and cumulative mapping functions are shown in Figs. 12(a)–12(c) and Figs. 12(d)–12(f), respectively, for frames 275, 300, and 315. Focusing on the blood vessel and specular highlight (blue arrows), we can see that as the anatomy is covered by the tool over a longer period of time in

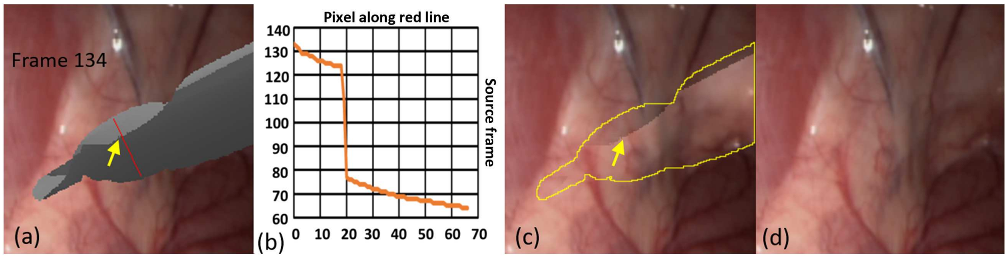


Fig. 11 Example showing the results of the modified Poisson blending algorithm: (a) gray scale corresponds to the source frame used to inpaint tool region; (b) plot of source frame used to inpaint tool region as a function of position along red line in panel (a); (c) inpainted results using Poisson blending algorithm; and (d) inpainted results using modified Poisson blending algorithm.

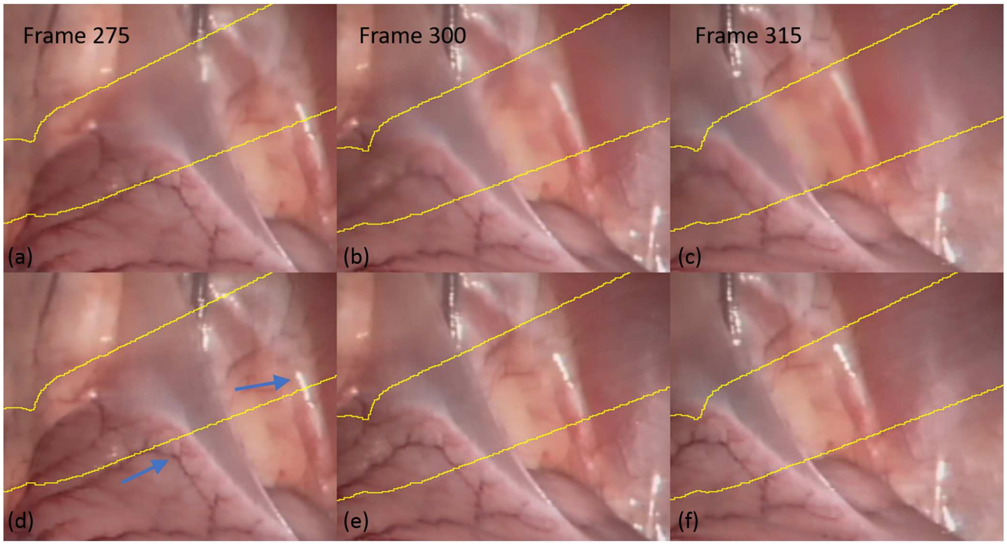


Fig. 12 Comparison of using a simple (noncumulative) versus cumulative mapping function to inpaint the tool region using a parametric optical flow model for frames 275, 300, and 315 where the anatomy under the tool is changing slowly: (a)–(c) noncumulative mapping function; (d)–(f) cumulative mapping function. Focusing on the specular highlight and blood vessel it can be seen that inpainting with the cumulative mapping function leads to sharper results.

successive frames, the resulting inpainted images become blurrier while the cumulative mapping results remain sharp.

For the straightforward mapping, the information used to inpaint the tool comes from the previous inpainted frame, and depending on how the tool is moving, either the inpainted tool region or background. For cumulative mapping, the pixel data used to inpaint the tool come from the source frame where the covered anatomy was last discernible in the background area (i.e., uncovered by the tool). Because the source pixels used to inpaint the tool region are now only copied once via interpolation from the source frame as opposed to the simple mapping where the source data may have been copied multiple times, the cumulative mapping eliminates the blurriness issue that occurs with the simple mapping.

3.4.2 *Tool removal results: method B*

In Fig. 13, we display the outcomes of tool removal method B using a non-parametric optical flow-based model to eliminate the tool from a video of a cardiac surface deforming as a result of both cardiac motion and respiration. The reference frames, which are 150 consecutive frames long and include multiple cycles of the deforming cardiac surface, are taken prior to the introduction of the surgical tools. The frame containing the tool is shown in Fig. 13(a), the inpainted results using the nearest reference frame that has been spatially transformed by an optical flow-based model are shown in Fig. 13(b), and the ground truth is shown in Fig. 13(c). It is clear that the reference image frame inpainting technique yields outcomes that are visually similar to the actual data. The specular highlights in the inpainted region serve as the primary visual distinction between the inpainted results and the ground truth. Given that the reference frame and the ground truth frame were taken at various times, the specular highlights in the images are not always consistent.

In Fig. 14, we show a comparison between copying and pasting the pixels of the closest reference frame before [Fig. 14(a)] and after applying the optical flow transformation [Fig. 14(b)] for inpainting using method B. Note in the figure that we refer to copy and paste to inpainting before and optical flow to inpainting after the applying the spatial transformation to the closest reference frame. Focusing on the region within the black circle [Fig. 14(a)], it can be seen that applying the optical flow transformation improves and generates inpainting results that are very similar to the ground truth. In many of the inpainted frames, the copy and paste method results when observing a single stimulus, which is observing one frame at a time, produce inpainted

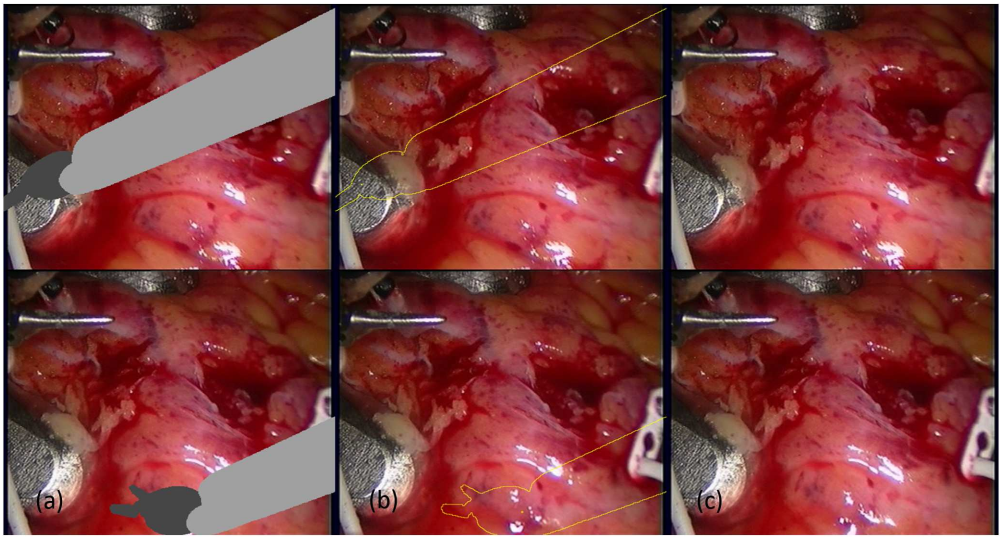


Fig. 13 Tool removal using method B with non-parametric optical flow-based model: (a) tool containing frames; (b) inpainted results using closest reference frame; and (c) ground truth frames (Video 2, MP4, 960 KB [URL: https://doi.org/10.1117/1.JMI.10.4.045002.s2]).

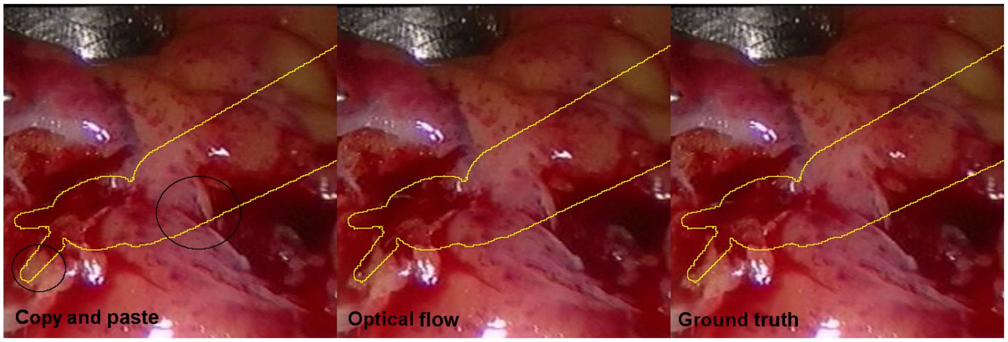


Fig. 14 Comparison between copying and pasting the pixels of the closest reference frame before and after applying the optical flow transformation for inpainting using method B.

results that visually look very acceptable, but when observed in a video playback becomes very obvious that the results are not accurate and are improved by the optical spatial transformation.

In Table 2, we report the quantitative evaluation of the inpainted videos using MSE, PSNR, and SSIM[40] image quality metrics. It can be noted that MSE and PSNR are not always well-correlated with perceived/subjective visual quality, whereas SSIM shows better correlations.

For the method A example, we provide a comparison between the inpainted and Poisson blended inpainted results as an illustration. In this case, the algorithm does a good job of selecting the proper pixels from earlier frames to fill in the occluded area. However, these image pixels come from previous frames whose illumination of the occluded anatomy was different from that of the present frame [see Fig. 5(b)]. As a result, the majority of the errors in this example are nonstructural errors, and they can be reduced by minimizing illumination mismatches using the Poisson blending algorithm.

We compare copying and pasting the closest reference frame's pixels before and after the optical flow transformation for the method B example. Since the camera is still in this instance, the illumination is essentially constant, despite variations in the specular highlights brought on by changes in the surface of the beating heart. Because the reference and current frames may not have been properly matched, the errors in this example are mostly structural. The insufficient frame rate of the video capture is most likely to blame for the absence of a matching frame. Although it is also well known that the underlying stochastic nature of the beating heart is partly a result of the stochastic characteristics of the ion channels,[41] it can never completely eliminate the structural errors, the spatial transformation can help to reduce these errors.

Table 2 Quantitative evaluation of the tool removal methods for synthetic tools in terms of MSE, PSNR, and SSIM.

	Metric		
Method	MSE (avg/min/max) (smaller better)	PSNR (avg/min/max) (larger better)	SSIM (avg/min/max) (larger better)
Method A: affine transformation (640 × 480 × 135)	690.9/58.0/2111.6	22.5/14.9/30.5	0.932/0.797/0.993
Method A: affine transformation with Poisson blending	41.5/6.5/163.9	33.3/26.0/40.0	0.993/0.958/0.999
Method B: copy and paste (720 × 576 × 500)	223.7/40.8/1183.5	25.4/17.4/32.0	0.971/0.937/0.994
Method B: optical flow warping	125.0/16.7/641.7	28.1/20.1/35.9	0.980/0.948/0.994

4 Discussion and Conclusion

To enable visualization of the anatomy that the tool was covering up, this research demonstrates a novel application of segmenting and digitally removing the surgical instruments from laparoscopic/endoscopic video.

We suggested a modified U-NetPlus for the segmentation of surgical instruments. We used a pre-trained model as the encoder with batch normalization, which converges much faster than the network trained from scratch, to increase robustness beyond that of the U-Net framework. We replaced the deconvolution layer in the decoder section with two convolution layers, followed by an upsampling layer that employs NN interpolation. To avoid the overfitting issue, we also used a quick and efficient data augmentation technique. Our evaluation was based on the MICCAI 2017 EndoVis results. Using the MICCAI 2017 EndoVis Challenge dataset, we assessed its performance. The results of our suggested model were also seen as standalone surgical instrument segmentation and as overlays on the original endoscopic images. In addition, we carried out an "attention study" to find out where our suggested algorithm "looks" in an image.

According to the Jaccard and DICE metrics, our proposed model with batch-normalized U-NetPlus-VGG-16 outperforms existing approaches. It achieved 90.20% DICE for binary class segmentation and 76.26% for parts segmentation, both of which demonstrated at least a 0.21% percent improvement over the existing approaches and a more than 6% improvement over the traditional U-Net architecture. However, U-NetPlus-VGG-16 performed worse than U-NetPlus-VGG-11 in terms of determining the instrument type while the other widely disseminated techniques performed marginally better. Our paired statistical test demonstrated a significant improvement over the performance of the TernausNet method, despite the fact that the improvement is still modest.

We performed the aforementioned paired statistical tests comparing the output of our proposed method and that of the other networks to assess the performance improvement in segmentation that was produced by our proposed method. The test resulted in a significant performance difference between the U-NetPlus framework and the TernausNet and U-Net architectures ($p < 0.05$). Despite the fact that many existing techniques and approaches use interpolation on an encoder-decoder network's upsampling path for various segmentation goals, a key component of our research is the skillful blending and adaptation of existing techniques to increase the segmentation accuracy of surgical instruments. In addition, we emphasize that our primary contribution is to enhance U-NetPlus by altering the TernausNet to lessen some of the artifacts that are still present. Therefore, even though this work does not offer a wholly original framework, it does show how carefully combining and integrating previous contributions can result in improved performance. Overall, the surgical tool binary segmentation achieved by our tool segmentation architecture is accurate enough to be trusted.

It should be noted that the da Vinci-labeled ground truth data does not always represent an exact segmentation of the surgical tool [see Figs. 15(b) and 15(d)]. Due to misalignments

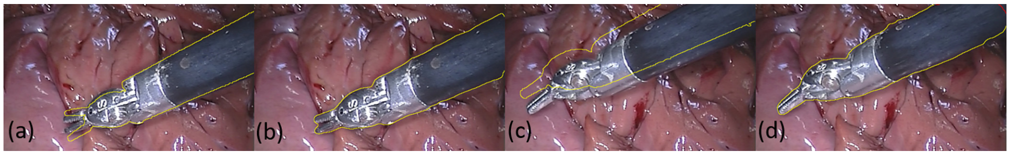


Fig. 15 Qualitative evaluation of segmentation results: (a) and (c) ground truth generated by forward kinematics of the da Vinci Research Kit; (b) and (d) segmentation results from our U-NetPlus segementor.

between the tool outline reconstructed from the forward kinematics of the da Vinci Research Kit and the actual tool appearance in the image frame, there are significant limitations that essentially cast doubt on the accuracy of the ground truth data. The tool outlines produced by our segmentation technique, however, are more precise than those produced from the ground truth forward kinematics [see Figs. 15(a) and 15(c)].

The instrument segmentation mask is filled in (or inpainted) by the tool removal algorithms using a tool segmentation mask and either previous instrument-containing frames or instrument-free reference frames. On a dataset of robotic instruments from the MICCAI 2015 EndoVis Challenge, we have shown how well the proposed surgical tool segmentation and removal algorithms perform. In addition, we demonstrated the tool removal algorithm's successful operation from surgical tool-free videos that contained videos of moving surgical tools that were generated artificially.

Our study shows that the proposed inpainting methods, together with the automated surgical instrument detection, classification, and segmentation tool, can effectively identify and mask the surgical instruments from endoscopic videos and render them translucent by inpainting them with background tissue information. As such, these proposed methods have the potential to enable surgeons to perform endoscopic-guided minimally invasive interventions with greater ease, without the need to constantly and repetitively retract the surgical instruments from the field-of-view to visualize the tissue otherwise occluded.

In conclusion, this work is the first to show how a modified U-Net decoder can be used to eliminate artifacts brought on by the transposed convolution using NN interpolation. Our suggested architecture is used to (1) segment the surgical tools from laparoscopic images, which performed better than the state-of-the-art TernausNet framework, and to (2) successfully remove the surgical tool, producing results that are visually comparable to the actual findings.

Disclosures

This paper is an extension of our previously published conference paper in SPIE Medical Imaging 2021. In this paper, we expanded our introduction and related works section, added more details in the methodology on the surgical instrument segmentation algorithm, presented additional results and analysis, and provided a more comprehensive discussion section. You can view the entire videos for our inpainting results at https://smkamrulhasan.github.io/.

Acknowledgments

Research reported in this publication was supported by the National Institute of General Medical Sciences of the National Institutes of Health (Award No. R35GM128877) and by the Office of Advanced Cyber infrastructure of the National Science Foundation (Award No. 1808530).

References

1. P. P. Rao, "Robotic surgery: new robots and finally some real competition!" *World J. Urol.* **36**(4), 537–541 (2018).
2. Z. Fu et al., "The future of endoscopic navigation: a review of advanced endoscopic vision technology," *IEEE Access* **9**, 41144–41167 (2021).
3. S. J. Spaner and G. L. Warnock, "A brief history of endoscopy, laparoscopy, and laparoscopic surgery," *J. Laparoendosc. Adv. Surg. Tech.* **7**(6), 369–373 (1997).
4. M. Kim et al., "Evolution of spinal endoscopic surgery," *Neurospine* **16**(1), 6 (2019).
5. D. Xie et al., "Surgical instruments hyalinization: occlusion removal in minimally invasive endoscopic surgery," *Biomimetic Intell. Robot.* **3**(3), 100105 (2023).

6. H. Mo et al., "Task autonomy of a flexible endoscopic system for laser-assisted surgery," *Cyborg. Bionic Syst.* **2022**, 1–11 (2022).
7. M. Yang et al., "DenseASPP for semantic segmentation in street scenes," in *Proc. IEEE Conf. Comput. Vis. and Pattern Recognit.*, pp. 3684–3692 (2018).
8. K. He et al., "Mask r-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2961–2969 (2017).
9. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. and Pattern Recognit.*, pp. 3431–3440 (2015).
10. R. Mechrez, J. Goldberger, and H. Greenspan, "Patch-based segmentation with spatial consistency: application to MS lesions in brain MRI," *J. Biomed. Imaging* **2016**, 7952541 (2016).
11. D. Shen, G. Wu, and H.-I. Suk, "Deep learning in medical image analysis," *Annu. Rev. Biomed. Eng.* **19**, 221–248 (2017).
12. O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," *Lect. Notes Comput. Sci.* **9351**, 234–241 (2015).
13. C. Chen et al., "Learning to see in the dark," arXiv:1805.01934 (2018).
14. N. Jiang and L. Wang, "Quantum image scaling using nearest neighbor interpolation," *Quantum Inf. Process.* **14**(5), 1559–1571 (2015).
15. X. Jia, H. Chang, and T. Tuytelaars, "Super-resolution with deep adaptive image resampling," arXiv:1712.06463 (2017).
16. A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill* **1**(10), e3 (2016).
17. L. C. Garca-Peraza-Herrera et al., "ToolNet: holistically-nested real-time segmentation of robotic surgical tools," in *IEEE/RSJ Int. Conf. Intell. Robot. and Syst. (IROS)*, IEEE, pp. 5717–5722 (2017).
18. A. Shvets et al., "Automatic instrument segmentation in robot-assisted surgery using deep learning," arXiv:1803.01207 (2018).
19. D. Pakhomov et al., "Deep residual learning for instrument segmentation in robotic surgery," arXiv:1703.08580 (2017).
20. A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv:1511.06434 (2015).
21. T. Salimans et al., "Improved techniques for training GANs," in *Adv. Neural Inf. Process. Syst.*, pp. 2234–2242 (2016).
22. V. Iglovikov and A. Shvets, "TernausNet: U-net with VGG11 encoder pre-trained on ImageNet for image segmentation," arXiv:1801.05746 (2018).
23. K. He, R. Girshick, and P. Dollár, "Rethinking ImageNet pre-training," arXiv:1811.08883 (2018).
24. Y. Koreeda et al., "Virtually transparent surgical instruments in endoscopic surgery with augmentation of obscured regions," *Int. J. Comput. Assist. Radiol. Surg.* **11**(10), 1927–1936 (2016).
25. S. K. Hasan and C. A. Linte, "U-Netplus: a modified encoder-decoder U-Net architecture for semantic and instance segmentation of surgical instruments from laparoscopic images," in *41st Annu. Int. Conf. of the IEEE Eng. in Med. and Biol. Soc. (EMBC)*, IEEE, pp. 7205–7211 (2019).
26. S. Kornblith, J. Shlens, and Q. V. Le, "Do better ImageNet models transfer better?" arXiv:1805.08974 (2018).
27. S. Santurkar et al., "How does batch normalization help optimization? (No, it is not about internal covariate shift)," arXiv:1805.11604 (2018).
28. C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," *Lect. Notes Comput. Sci.* **9906**, 391–407 (2016).
29. A. Bokov and D. Vatolin, "100+ times faster video completion by optical-flow-guided variational refinement," in *25th IEEE Int. Conf. Image Process. (ICIP)*, IEEE, pp. 2122–2126 (2018).
30. A. Bokov and D. Vatolin, "Toward efficient background reconstruction for 3D-view synthesis in dynamic scenes," in *IEEE Int. Conf. on Multimedia & Expo Workshops (ICMEW)*, IEEE, pp. 37–42 (2017).
31. S. K. Hasan, R. A. Simon, and C. A. Linte, "Segmentation and removal of surgical instruments for background scene visualization from endoscopic/laparoscopic video," *Proc. SPIE* **11598**, 55–61 (2021).
32. S. Baker and I. Matthews, "Lucas-Kanade 20 years on: a unifying framework," *Int. J. Comput. Vis.* **56**(3), 221–255 (2004).
33. E. Meinhardt-Llopis, J. S. Pérez, and D. Kondermann, "Horn-Schunck optical flow with a multi-scale strategy," *Image Process. On Line* **3**, 151–172 (2013).
34. P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," in *ACM SIGGRAPH 2003 Papers*, pp. 313–318 (2003).
35. "MICCAI 2017 endoscopic vision challenge: robotic instrument segmentation sub-challenge," (2017). https://endovissub2017-roboticinstrumentsegmentation.grand-challenge.org/Data/.
36. M. Ye et al., "Self-supervised Siamese learning on stereo image pairs for depth estimation in robotic surgery," arXiv:1705.08260 (2017).
37. A. Buslaev et al., "Albumentations: fast and flexible image augmentations," *Information* **11**(2), 125 (2020).

38. I. Laina et al., "Concurrent segmentation and localization for tracking of surgical instruments," *Lect. Notes Comput. Sci.* **10434**, 664–672 (2017).
39. R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," arXiv:1704.03296 (2017).
40. T. Samajdar and M. I. Quraishi, "Analysis and evaluation of image quality metrics," in *Information Systems Design and Intelligent Applications*, J. Mandal et al., Eds., pp. 369–378, Springer (2015).
41. Z. Qu et al., "Nonlinear and stochastic dynamics in the heart," *Phys. Rep.* **543**(2), 61–162 (2014).

S. M. Kamrul Hasan completed his PhD at Chester F. Carlson Center for Imaging Science at Rochester Institute of Technology (RIT), Rochester, New York. He worked at Philips Research in Cambridge, Massachusetts, and IBM Research in California, as a machine learning research intern, in 2021 and 2020 respectively. His research focuses broadly on developing and optimizing machine learning models for analyzing multi-modal images to enable more accurate automatic semantic and instance segmentation.

Richard A. Simon is currently working as a research scientist in the Department of Biomedical Engineering, Rochester Institute of Technology, New York.

Cristian A. Linte completed his PhD in biomedical engineering at the University of Western Ontario in Ontario, Canada, in 2010. In 2011, he joined the biomedical imaging resource at Mayo Clinic in Rochester, Minnesota – a group with a long-standing tradition in the development of medical image analysis and image-guided intervention technology – for a postdoctoral fellowship, which transitioned into an research-track academic appointment as assistant professor of biomedical engineering in 2012.