Robotic instrument segmentation with image-to-image translation

Emanuele Colleoni[1], Danail Stoyanov[1]

***Abstract*—The semantic segmentation of robotic surgery video and the delineation of robotic instruments are important for enabling automation. Despite major recent progresses, the majority of the latest deep learning models for instrument detection and segmentation rely on large datasets with ground truth labels. While demonstrating the capability, reliance on large labelled data is a problem for practical applications because systems would need to be re-trained on domain variations such as procedure type or instrument sets. In this paper, we propose to alleviate this problem by training deep learning models on datasets that are synthesised using image-to-image translation techniques and we investigate different methods to perform this process optimally. Experimentally, we demonstrate that the same deep network architecture for robotic instrument segmentation can be trained on both real data and on our proposed synthetic data without affecting the quality of the output models' performance. We show this for several recent approaches and provide experimental support on publicly available datasets, which highlight the potential value of this approach.**

***Index Terms*—Medical Robots and Systems, Deep Learning Methods, Image-to-Image Translation, Surgical Robot Simulators, Surgical tool segmentation.**

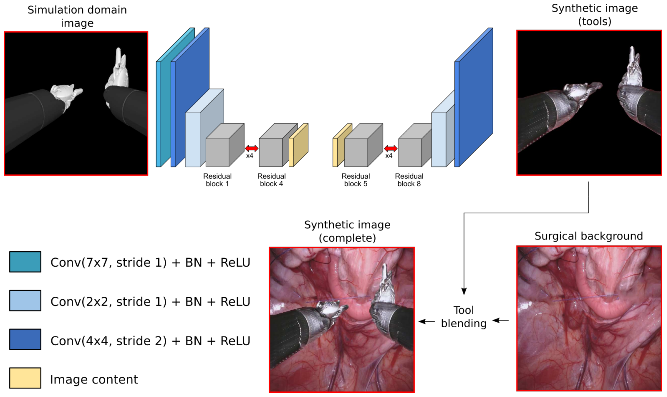


Fig. 1: Surgical workflow of the proposed method. Synthetic frames with segmentation labels are produced using image to image translation on simulated instruments to transform them into realistic surgical tools. These are finally blended on a surgical background to obtain full synthetic frames.

I. Introduction

COMPUTER Assisted Interventions (CAI) can support enhanced capabilities in robotic surgery by enabling cognitive functions in conjunction to the articulated instrumentation. Such CAI systems may include effective transfer of preoperative planning to the intra-operative surgical site, critical structure localization and preservation, and even autonomous task execution [1], [2]. Knowing the location and pose of instruments during surgery is a fundamental building block for such systems and has been demonstrated as practical using vision algorithms on endoscopic video [3]. Labelling, however, remains a challenge for supervised deep learning vision models in surgical data [2] .

The reliability and robustness of tool detection and segmentation using vision has improved significantly with deep learning advances [3]. Recently, solutions to alleviate the need for supervisory labels in surgical video models have been proposed utilizing surgical simulation and image-to-image translation (I2I) techniques to produce synthetic surgical data [4], [5]. Simulation data can generate reliable training labels but has significant limitations in reproducing realistic photometric style features such as colors, texture and illumination conditions. Yet the domain gap between simulation and real endoscopic images could be bridged because recent advances in I2I allow the transfer of style features from different image domains without the need for paired-samples, thus making possible to transfer the realistic style from real surgical frames to simulation ones, where labels are provided automatically [6].

In this paper, we develop the I2I paradigm for training surgical instrument detection and segmentation models. Our method generates simulation images of robotic instruments and we perform a comparative analysis of different I2I style feature transfer approaches to determine optimal performance. We then experimentally show that the synthesised data can be used to train vision models with performance equivalent to those trained on real data and we investigate how image and segmentation quality metrics relate to optimize our results. Importantly, we show that by training the same network architecture using synthetic and real data produces no discernible difference, which highlights a promising advance towards avoiding supervised learning in surgical applications.

II. RELATED WORK

Deep neural architectures for stylized image synthesis from features extracted at different depths of source data domains [7] have received intense interest in recent years [8]. Various

Manuscript received: October, 15, 2020; Revised December, 30, 2020; Accepted January, 22, 2021.

This paper was recommended for publication by Editor Pietro Valdastri upon evaluation of the Associate Editor and Reviewers' comments.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

[1]E. Colleoni and D. Stoyanov are with the Wellcome/EPSRC Centre for Interventional and Surgical Sciences (WEISS), University College London (UCL), UK emanuele.colleoni.19@ucl.ac.uk

Digital Object Identifier (DOI): see top of this page.

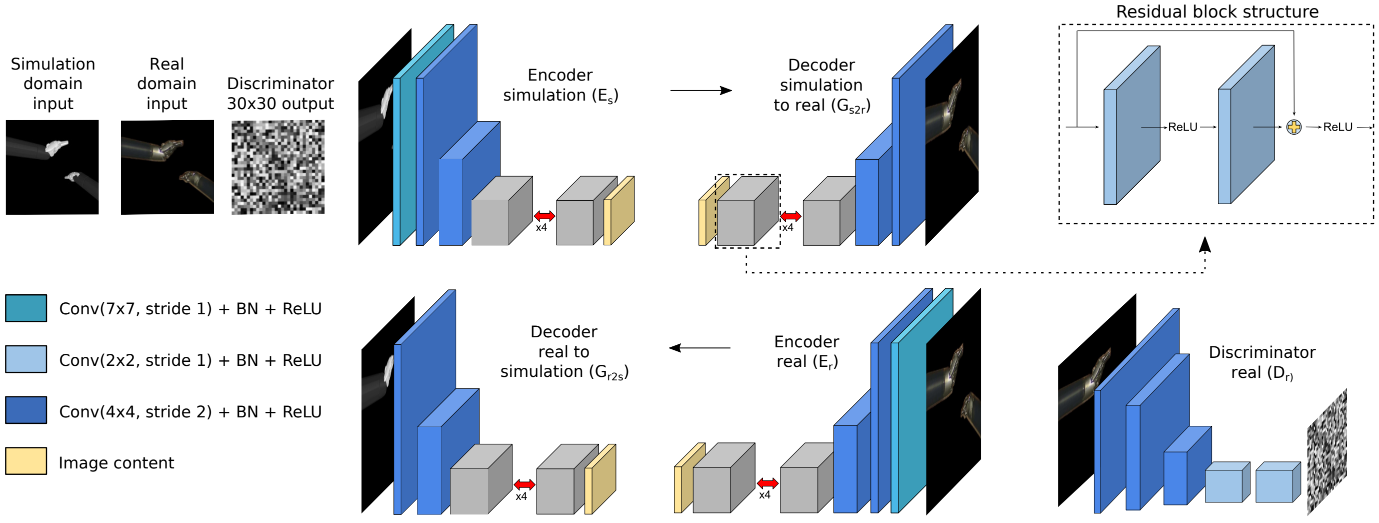

Simulation domain input
Real domain input
Discriminator 30x30 output
Encoder simulation (E_s)
Decoder simulation to real (G_s2r)
Residual block structure
ReLU
ReLU
x4
Conv(7x7, stride 1) + BN + ReLU
Conv(2x2, stride 1) + BN + ReLU
Conv(4x4, stride 2) + BN + ReLU
Image content
Decoder real to simulation (G_r2s)
Encoder real (E_r)
Discriminator real (D_r)

Fig. 2: Architecture of the I2I model employed in the proposed workflow. Only one discriminator is illustrated for simplicity.

efforts have focused on advancing style transfer with photorealistic improvements, computational efficiency and synthesis outside the training domain. In surgical applications, style transfer from real to simulation images in retinal surgery images has been reported [9] but still requires accurate labelling of all represented structures in the input space to learn a mapping from one image to another. Major advances in avoiding labelling have been achieved with generative adversarial networks (GANs) [10], [11] and in particular for realistic image synthesis with cycle-GANs [12]. Image content and style loss functions based on the features extracted by each encoder during image translation have been shown to be particularly effective to reduce the image artifacts introduced by GANs and to enforce the content to be consistent [13] and we utilize this paradigm in the proposed method.

The use of cycle-GANs in surgical data generation for I2I techniques has been shown with a self-attention mechanism and a surgical tool classifier to move between different surgical tools [5]. Similar approaches have investigated how to produce synthetic in-vivo frames from cadaver images [14] and I2I methodologies for phantom to real image style transfer [15]. For simulation to real transfer, which is most relevant to our method, a Multimodal Unsupervised Image-to-Image Translation (MUNIT)-based framework for I2I for liver segmentation has recently been shown [4]. A Structural Similarity Index Measure (*SSIM*) [16] based loss was used to improve image quality, but despite high quality results in tissue renditions, the style of the surgical tools was often corrupted during the translation, showing features typical of the image background, such as vessels and specularities. More recently, a simulation to real I2I framework for laparoscopic image synthesis has also been reported but without investigating multiple loss combinations nor experiments showing that segmentation models can be trained solely on synthetic frames [17].

III. Methods

The overall workflow we proposed for surgical dataset generation is presented in Fig. 1. As first fundamental step, a trained I2I model is required to transform images from simulation into real domain. A descriptions of the employed network architectures and loss functions are given in Sec. III-A and Sec. III-B respectively. In Sec. III-C we present the structure of the domain sets (i.e. simulation and ex-vivo/in-vivo domains) and their generation procedure. As second step, we produce each synthetic frame by first transforming an image acquired from the robot simulator to have realistic style and then we blend it onto a surgical background. In Sec. III-D we report both those stages in details. (N.B.: In this section, the terms 'ex-vivo or in-vivo images' and 'real images' are used as synonyms.)

A. Network architecture

The I2I model architecture we used in our work is presented in Fig. 2. Following cycle-GAN and MUNIT frameworks [12], [13], the network is composed by two discriminators D_s, D_r (subscripts s and r stands for *simulation* and *real* respectively) and two generators.

Each generator is in turn composed by an encoding part (e.g. E_s, E_r) and a decoding part (G_{s2r}, G_{r2s}). Encoders work as feature extractors and their output is a series of image features that characterizes the image content (c_s / c_r). On the other hand, decoders are associated with a particular style defined by the image domain they have to reconstruct and work to produce domain images starting from encoder's outputs. In formulas: $c_s = E_s(I_s)$ and $I_{s2r} = G_{s2r}(c_s)$, where I_s is a sample image from the simulation domain, c_s are the image features extracted by the simulation encoder and I_{s2r} is the transformed image. The opposite transformation can be obtained by using E_r and G_{r2s} on images from ex-vivo/in-vivo domain (I_r).

Images are processed into the encoders through several convolutional blocks. As common in famous architectures like ResNet [18], a first 7x7 convolution layer first processes the input without modifying the image width and height but increasing the number of channels to 64. This is followed by two down-sampling blocks, each performing a 4x4 convolution with stride 2 that halves the image dimensions and doubles the number of channels. The result is finally processed through 4 consecutive residual blocks, each one composed by two convolutional layers, where the second output is concatenated with the block's input using a skip connection.

Decoders architecture is specular to the encoders one, although the initial 7x7 convolution is not replicated and down-sampling blocks are substituted with up-sampling ones to recover the original image shape. Finally, we implemented the discriminators following PatchGAN framework [19]: images are processed through 3 consecutive 4x4 down-sampling blocks (stride=2) followed by 2 convolutional blocks (3x3, stride=1), producing 30x30 grayscale images as output. Each pixel evaluates a 70x70 patch of the input image and represents the probability for that patch to be real or fake. Each convolution in all the described architectures is followed by spectral normalization [20], Instance Normalization (*IN*) [21] and leaky rectified linear unit activation function (leaky-ReLU), except for output layers of each module, where *IN* is not applied. All these techniques have been shown to be particularly effective to improve generative models image quality and to stabilize their training [20], [21].

B. Loss functions

In this work we considered 4 loss functions that has shown to be extremely effective for high quality I2I:

1) Adversarial loss: The adversarial loss is the fundamental building block at the core of GANs and is responsible for the translation of domain-specific features. For a more detailed description, please refer to [22], [10]. In this work we use the least square adversarial loss [22], a variation of the original formulation [10] that has shown to produce better quality images.

Let $I_r \in \chi_r$ and $I_{s2r} \in \chi_{s2r}$ denote samples from the ex-vivo/in-vivo images distribution χ_r and the distribution of synthetic images χ_{s2r} respectively. The real discriminator D_r adversarial loss is defined as:

$$\begin{aligned} L^{D_r}_{GAN}(D_r) = \frac{1}{2}\mathbb{E}_{I_r}[(D_r(I_r)-1)^2]+ \\ +\frac{1}{2}\mathbb{E}_{I_{s2r}}[(D_r(I_{s2r}))^2], \end{aligned} \tag{1}$$

while the adversarial loss related to the simulation encoder E_s and to the simulation-to-real generator G_{s2r} is:

$$L^{G_r}_{GAN}(E_s,\ G_{s2r}) = \frac{1}{2}\mathbb{E}_{I_{s2r}}[(D_r(I_{s2r})-1)^2] \tag{2}$$

The simulation losses $L^{D_s}_{GAN}(D_s)$ and $L^{G_s}_{GAN}(G_{r2s})$ can be obtained using the same formula with the distributions $I_s \in \chi_s$ and $I_{r2s} \in \chi_{r2s}$.

2) Cycle consistency loss: Along with the other loss functions described below, the cycle consistency loss [12] is responsible for the preservation of content features during the image translation process. This loss is defined as:

$$\begin{aligned} L_{cyc}(E_s,\ E_r,\ G_{s2r},\ G_{r2s}) = \mathbb{E}_{I_{s2r2s}}\|I_{s2r2s} - I_s\|_1+ \\ +\mathbb{E}_{I_{r2s2r}}\|I_{r2s2r} - I_r\|_1, \end{aligned} \tag{3}$$

where I_{s2r2s} is the same simulation image I_s after being transformed into the real domain $I_{s2r} = G_{s2r}(c_s)$ and then transformed back to the simulation one $I_{s2r2s} = G_{r2s}(c_{s2r})$, while I_{r2s2r} was obtained with the inverse cycle transformation. Such formulation forces the generators to prioritize transformations of the image style while discouraging severe modifications in the content.

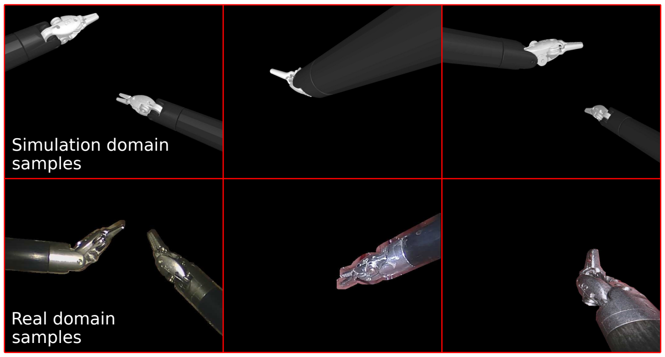


Fig. 3: Sample frames from simulation and real domain sets.

3) Content loss: As described in MUNIT framework [13], the content loss is based on the assumption that both the considered domains share the same content features while their style is different and domain dependent. Thus, the content of an image from the simulation domain c_s and the content of the transformed image c_{s2r} should be equal (and vice-versa with c_r and c_{r2s}). From this hypothesis, X. Huang et al. formulated the content loss as:

$$\begin{aligned} L_{con}(E_s,\ E_r,\ G_{s2r},\ G_{r2s}) = \\ \mathbb{E}_{c_s}\|E_r(G_{s2r}(c_s)) - c_s\|_1+ \\ +\mathbb{E}_{c_r}\|E_s(G_{r2s}(c_r)) - c_r\|_1 \end{aligned} \tag{4}$$

In [13], the authors proposed a style loss in addition to the content loss to perform multimodal I2I. However, since we are investigating a unimodal approach, we chose to not consider the style loss in our formulation. For this reason, the considered MUNIT framework will be referred as MUNIT*.

4) Structure similarity loss: Following [4], we chose the structure similarity score (*SSIM*) $[0 \leq SSIM \leq 2]$ as our final loss. Details about the formulation of *SSIM* score can be found in [16]. Since better performances are achieved when *SSIM* score approaches 2, the associated loss requires the negative form to be minimized during training.

The *SSIM* based loss is defined as:

$$L^{s}_{SSIM}(E_s,\ E_r,\ G_{s2r},\ G_{r2s}) = 2 - SSIM(I_{s2r2s}, I_s) \tag{5}$$

Again, the *SSIM* loss in the inverse direction L^{r}_{SSIM} can be obtained by substituting I_{r2s2r} and I_r in (5).

5) Overall loss: The final formulation for the loss we employed to train our I2I model is reported below:

$$\begin{aligned} L(E_s,\ E_r,\ G_{s2r},\ G_{r2s},\ D_s,\ D_r) = \\ \lambda_{\text{adv}} * (L^{D_r}_{GAN} + L^{D_s}_{GAN} + L^{G_r}_{GAN} + L^{G_s}_{GAN})+ \\ \lambda_{\text{cyc}} * L_{cyc} + \lambda_{\text{con}} * L_{con}+ \\ \lambda_{SSIM} * (L^{s}_{SSIM} + L^{r}_{SSIM}) \end{aligned} \tag{6}$$

where $\lambda_{\text{adv}}, \lambda_{\text{cyc}}, \lambda_{\text{con}}, \lambda_{SSIM}$ are weight hyperparameters to scale each loss module individually.

C. Domain sets: generation and pre-processing

1) Real domain: The real domain set was built using video frames from pre-existing ex-vivo and in-vivo datasets along with their segmentation ground truth. Each frame was pre-processed using its binary segmentation mask to set all the background pixels to 0, thus extracting the tools from their context. We performed an ablation study where we trained the I2I network using real domain frames with no segmentation pre-processing, showing that the model was unable to learn the translation between simulation and real domains. Images showing the results of this experiment can be found in the additional materials. Sample images from both real and simulation sets are presented in Fig. 3.

2) Simulation domain: To produce the simulation domain set, we employed the CoppeliaSim[1] daVinci virtual simulator developed in [23]. Depending on the availability of kinematic data associated to real domain frames, we used a MATLAB[2]/ROS[3] interface to first position the tools in the camera field of view (FOV) and then we moved the tools according to the kinematic data stream, collecting frames at a fixed rate. When no kinematic data were provided, we simply randomized the tools position by adding uniform noise in each joint of the da Vinci simulator. The procedure was performed in loop and a video frame was collected at each step. This technique allowed us to automatically collect labelled simulation frames with high inter-variability. Since this work is mainly focused on surgical tool style translation, we did not use any background, leaving non-tool pixels set to 0. Our datasets include only EndoWrist® Large Needle Drivers, however, the procedure could be translated to any minimally invasive surgery instrument.

D. Surgical background production and tool blending

Once the network was trained and simulation frames could be transformed to have a realistic style, as shown in Fig. 1, we blended them onto surgical background images to produce complete surgical frames. Those background images were generated starting from authentic frames by removing the surgical instruments with the image inpainting tool developed in [24]. Samples of generated background images can be found in the additional materials. The synthetic tools were blended onto the surgical background using the following formula:

$$I_{full} = I_{tools} * M + I_{background} * (1 - M) \tag{7}$$

where I_{full} is the final synthetic frame, I_{tools} is the image containing the transformed tools, $I_{background}$ is the surgical background and M is the binary segmentation mask of the simulation frame used as network input to produce I_{tools}. An example of full synthetic frames is shown in Fig. 4.b.

IV. EXPERIMENTS

A. Datasets

In this work we considered 2 ex-vivo and 1 in-vivo datasets to perform our experiments:

1) MICCAI 2015 EndoVis challenge (ex-vivo): The MICCAI 2015 segmentation dataset[4] comprises 4 training videos of 45*sec* each and 6 additional videos for testing (4x15*sec* + 2x60*sec*), for a total of 8975 labelled frames.

2) MICCAI 2017 EndoVis challenge (in-vivo): From MICCAI 2017 dataset[5] we considered only the videos with EndoWrist® Large Needle Drivers in the FOV, with an overall amount of 4 train videos (1500 frames) and 2 test videos (150 frames), for a total of 4200 frames.

3) MICCAI 2020 from [6] (ex-vivo): This dataset is composed of 14 videos (10 for training/validation and 4 for test) and each one consists of 300 labelled frames. Frames from MICCAI 2020 come with associated kinematic data.

B. Experimental protocol

*1) Image-to-image translation models comparison (**E1**):* We first examined the performances of 4 different I2I frameworks on the considered datasets to investigate which loss function best suits the proposed task. All the considered models share the same architecture, but they vary the loss formulation. Specifically, we selected cycle-GAN ($\lambda_{\text{adv}} = 1, \lambda_{\text{cyc}} = 10, \lambda_{\text{con}} = 0, \lambda_{SSIM} = 0$), MUNIT* ($\lambda_{\text{adv}} = 1, \lambda_{\text{cyc}} = 10, \lambda_{\text{con}} = 10, \lambda_{SSIM} = 0$), cycle-GAN+*SSIM* ($\lambda_{\text{adv}} = 1, \lambda_{\text{cyc}} = 10, \lambda_{\text{con}} = 0, \lambda_{SSIM} = 10$) and MUNIT*+*SSIM* ($\lambda_{\text{adv}} = 1, \lambda_{\text{cyc}} = 10, \lambda_{\text{con}} = 10, \lambda_{SSIM} = 10$). Cycle and *SSIM* loss weights were chosen following [12] and [4] respectively. Content loss weight was set 10 times higher than in [13] since we experimentally observed that using λ_{con}=1 as in the original framework led the training loss into local minima, where I_{s2r} was completely black.

We trained each of these models on three pairs of real and synthetic domain sets separately, for a total of 12 training procedures. We built the three real domain sets by reshaping frames from each dataset (only training images were considered) to 720x576 and cropping them centrally with a 576x576 window. Then we selected frames removing images with movement artifacts and interlacing noise and applying the pre-processing procedure described in Sec. III-C1 to all the remaining frames. Overall, we selected 965, 915 and 420 frames from MICCAI 2015, 2017 and 2020 respectively. We produced the associated simulation domain sets using kinematic data from [6] for MICCAI 2020, while we randomised the tools position for the remaining two datasets, as described in Sec. III-C2.

During each training procedure, every 10 epochs, we performed style transfer on a subset of 500 simulation frames from the simulation domain sets and we quantitatively evaluated the visual performances of the results.

*2) Real vs synthetic datasets for surgical tool segmentation (**E2**):* From ***E1***, we selected the best models for each dataset to build our synthetic segmentation sets (Sec. III-D), thus creating 3 datasets (i.e. MICCAI 2015$_{syn}$, MICCAI 2017$_{syn}$, MICCAI 2020$_{syn}$). Each synthetic dataset was produced using surgical backgrounds from their correspondent

[1] https://www.coppeliarobotics.com/

[2] https://uk.mathworks.com

[3] https://www.ros.org/

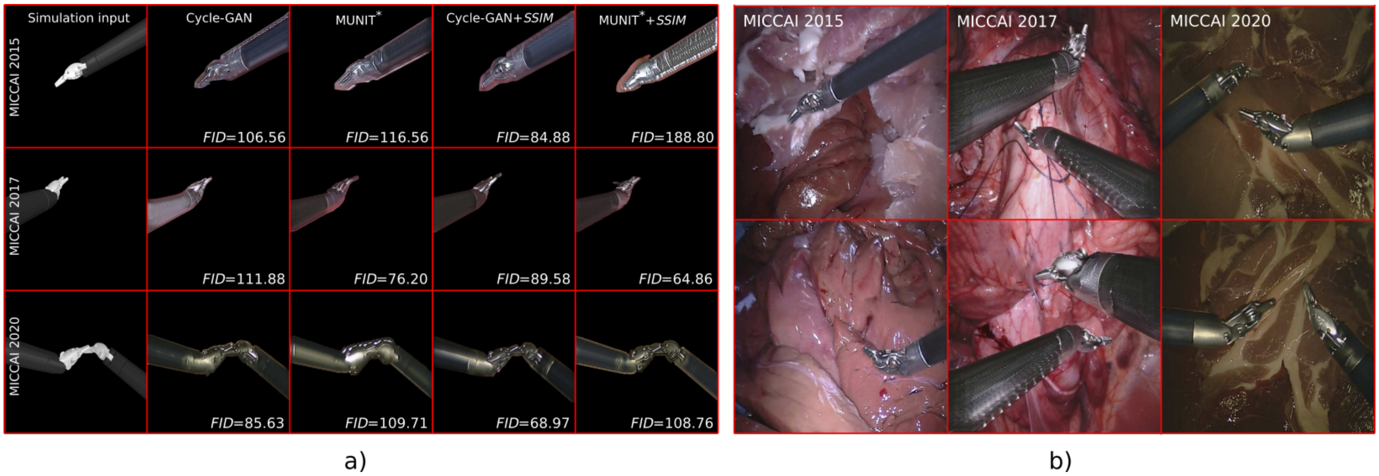
[4] https://endovissub-instrument.grand-challenge.org/

[5] https://endovissub2017-roboticinstrumentsegmentation.grand-challenge.org/

TABLE I

Quantitative results for the considered I2I models. The results were presented in terms of Frechet Inception Distance (*FID*) every 10 epochs. We highlighted in blue the best values for each epoch and in red the best model for each dataset.

		Epoch 10	***Epoch 20***	***Epoch 30***	***Epoch 40***	***Epoch 50***	***Epoch 60***	***Epoch 70***	***Epoch 80***	***Epoch 90***	***Epoch 100***
MICCAI 2015	***cycle-GAN***	143.96	215.70	110.84	120.35	106.56	264.74	97.18	88.16	96.82	91.40
	MUNIT*	207.32	203.28	192.16	131.47	116.56	121.87	118.13	116.07	101.12	95.70
	cycle-GAN+SSIM	104.71	124.71	131.61	96.46	84.88	104.03	137.52	90.92	101.45	87.28
	MUNIT*+SSIM	299.31	148.99	133.25	116.23	188.80	107.09	201.55	116.07	110.99	105.29
MICCAI 2017	***cycle-GAN***	125.17	81.25	95.11	114.08	111.88	85.51	89.86	87.94	97.88	74.35
	MUNIT*	195.34	110.43	94.34	80.15	76.20	82.30	97.33	80.91	66.04	68.23
	cycle-GAN+SSIM	105.16	84.15	83.77	67.17	89.58	87.24	77.80	74.80	84.51	66.43
	MUNIT*+SSIM	95.40	89.40	74.11	73.59	64.86	66.15	78.38	85.14	77.66	71.19
MICCAI 2020	***cycle-GAN***	120.74	92.76	78.68	92.04	85.63	78.211	86.59	69.47	77.73	69.05
	MUNIT*	172.87	203.93	128.74	119.43	109.71	101.68	103.58	109.94	104.31	98.73
	cycle-GAN+SSIM	101.19	113.23	79.45	70.26	68.97	91.89	96.27	72.00	66.25	60.15
	MUNIT*+SSIM	133.85	119.31	87.82	101.78	108.76	89.57	87.28	63.52	66.99	68.49



a)

b)

Fig. 4: Qualitative results from experiment ***E1***. In a), sample images from simulation are translated into the real domain using models trained with different loss functions (Sec. III-B) for 50 epochs. An example is reported for each dataset and loss. In b) we show full synthetic frames samples created using the best selected I2I models for each dataset.

real training set and with the same amount of training frames for fair comparison. Again, MICCAI 2020_{syn} was produced using kinematic data from [6] as simulation input, while frames for MICCAI 2015_{syn} and MICCAI 2017_{syn} were produced using random tool poses (N.B.: from this point on, we will refer to the dataset described in Sec. IV-A as 'real' sets, while addressing the datasets produced with our method as 'synthetic').

We chose a Unet pre-trained on Imagenet (ResNet34 [18] backbone) as our segmentation model and we trained it on each of the 3 real and synthetic datasets separately. For the real sets, we left the last 10% of each training video as validation set, while for the synthetic ones we produced the same number of validation images using our framework.

The segmentation performances of the trained models were evaluated on the test sets of each corresponding real dataset. Since in the second test video of MICCAI 2017 there is a EndoWrist® Fenestrated Grasper in the FOV (see Fig. 6, 5^{th} column) that was not present in the training synthetic frames but it was in the real ones, we did not consider the pixels in the neighborhood of the tool for fair comparison in calculating *IoU* score. Finally we performed a one-way ANOVA test on the results to assess their statistical significance.

C. Performance evaluation metrics

Quantitatively evaluating generative models performances in absence of paired samples is a difficult task that has been deeply studied over the last few years. In particular, Frechet Inception Distance (*FID*) has been widely employed as a metric to measure images quality [11] and it is defined as:

$$FID^2 = \|\mu_1 - \mu_2\|_2^2 + Tr(C_1 + C_2 - 2\sqrt{C_1 C_2}) \qquad (8)$$

where μ_1, μ_2 and C_1, C_2 are the mean values and covariances of two gaussians $G_1(\mu_1,\ C_1)$, $G_2(\mu_2,\ C_2)$ fitted on features extracted by a set of real and synthetic images respectively using an InceptionV3 network trained on Imagenet.

Although there is no evidence that this metric could be used to evaluate non-Imagenet images, several works showed that *FID* well correlates with human quality assessment even on these domains [25], [26], [27]. Following these paradigms and considering that, at the best of our knowledge, there is not a preferred or better method to evaluate generative models quality [28], we chose *FID* as our evaluation metric for ***E1***. We chose a sub-set of 500 images from the real training set of each dataset along with the synthetic frames produced in ***E1*** to calculate *FID* (the lower, the better) associated to each I2I model.

We evaluated the segmentation performances in ***E2*** using Intersection over Union score (*IoU*), that is one of the most employed metrics for this task [3]. *IoU* score is defined as:

$$IoU = \frac{TP}{TP + FP + FN} \tag{9}$$

where, for each image, *TP* is the number of pixels correctly labelled as tools, while *FP* and *FN* are the pixels misclassified as tools and background respectively.

Since we have no guarantees that a higher synthetic dataset quality (low *FID*) lead to improved segmentation performances, we trained a segmentation model for each MICCAI 2020 dataset produced using I2I models from ***E1*** (see Table I. We considered only MICCAI 2020 models at different epochs for resources constraints, 40 models overall) and we plotted the *IoU* score obtained by each model on MICCAI 2020$_{real}$ test set as a function of the *FID* score of its training dataset. Moreover, we trained a segmentation model on a dataset produced using simulation tools without I2I. Results are shown in Fig. 5, where each dot represents a different segmentation model. We performed least-squares regression on the results to highlight their relationship: the plot shows high negative linear correlation (r_value=-0.71, p_value<0.001, Wald test with α=0.05) [29] between the two considered scores. Moreover, most of the top *IoU* scores were achieved by models trained on datasets with *FID* score below 65. This suggests that *FID*, at least for lowest values, can be considered as a valid metric to choose the best I2I model. However, we acknowledge that this may not be the best evaluation metric, as will be discussed in Sec. VI-C. The red dot in Fig. 5 shows the segmentation performances of a model trained on a dataset produced with no I2I. The achieved *IoU* score below 40% shows the benefit introduced by the proposed workflow.

D. Implementation details

All the models discussed in this work were implemented in Tensorflow/Keras[6] and were trained on a GPU NVIDIA® Tesla® V100. In ***E1*** we employed Adam optimizer with learning rate α=0.0002 and linear decay of 4% at each epoch starting from epoch 75, for an overall of 100 epochs. We used a batch size of 1 and, following [11], we used historical averaging to stabilize the adversarial training.

For segmentation models training in ***E2*** we maintained Adam as our optimizer with default parameters, a batch size of 8 frames and binary cross-entropy + IoU score as loss function [6]. Each model was trained for 200 epochs and the best model for each dataset was selected as the one that achieved the best performances on the validation set.

V. Results

A. *E1* results

The quantitative results achieved in ***E1*** are presented in Table I. Cycle-GAN+*SSIM* model obtained the best performances in the majority of the considered training epochs for all the datasets, achieving the best overall *FID* scores in MICCAI 2015 and MICCAI 2020. MUNIT*+*SSIM* outperformed all the other models in MICCAI 2017 dataset, but performed poorly on all the remaining two, compared to all the other methods. Finally, cycle-GAN model obtained competitive results on both MICCAI 2015 and MICCAI 2020, never reaching however the best scores for any of them, while MUNIT* framework obtained the worst performances among all the considered sets. In Fig. 4.a we present visual results of each I2I model (rows 2-5) at epoch 50. Given these results, we selected cycle-GAN+*SSIM* as our model to produce the synthetic dataset.

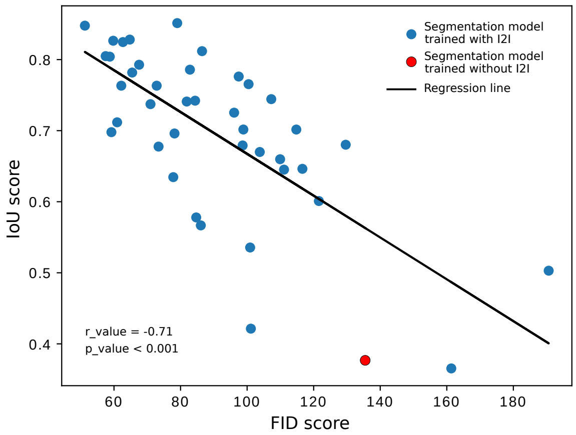


Fig. 5: Blue dots in the scatter plot represent the *IoU* scores achieved by different segmentation models on MICCAI 2020 test set as a function of the *FID* score (the lower, the better) of the synthetic training set. The red dot shows *IoU* and *FID* scores of a segmentation model trained on a dataset produced without I2I.

B. *E2* results

The quantitative results achieved by the segmentation model trained on synthetic and real frames are shown in Fig. 7. Overall, the model trained on real data outperformed the one trained on synthetic images in MICCAI 2015 and MICCAI 2020 datsets, with median *IoU* values of 81% and 96% against 77% and 92%, respectively. On the other hand, the use of a synthetic dataset showed training improvements for the model in MICCAI 2020 dataset, with an *IoU* score of 88%. All the results from synthetic and real sides showed to be statistically independent from each other, with an ANOVA test *p*-value<0.001 for all the datasets, as shown in the additional materials.

VI. Discussion

A. *E1* discussion

In general, the qualitative results presented in Fig. 4 reveal dependency between *FID* score and human quality perception. An example is shown for MUNIT*+*SSIM*, MICCAI 2015 (upper row, 5th column) and MUNIT*, MICCAI 2020 (lower row, 3rd column) frames, where I2I introduced pattern-like

[6] https://www.tensorflow.org/

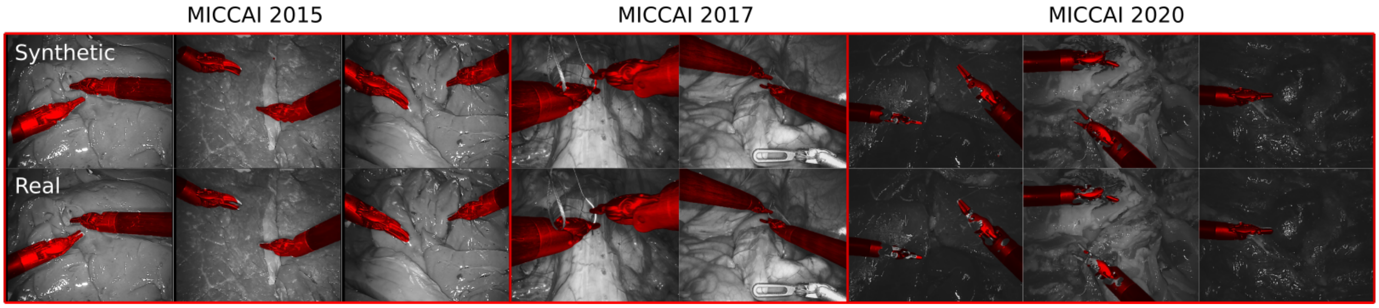


Fig. 6: Sample segmentation masks from models trained on synthetic and real frames on the considered test datasets.

artifacts on the shaft and a general loss of details, compared to simulation inputs (1^{st} column). These frames were labelled with an *FID* higher than all the others in the same rows, suggesting that this metric can successfully evaluate the quality of synthetic images.

The presence of the content loss during training showed a deterioration of the performances in both MICCAI 2015 and MICCAI 2017 datasets, thus surprisingly achieving best results in MICCAI 2017 with MUNIT*+*SSIM*. However, this could be explained by the presence of *SSIM* loss: as shown in Table I, cycle-GAN+*SSIM* and MUNIT*+*SSIM* achieved the majority of best scores in each dataset. This confirms the intuition of the authors in [4] that *SSIM* loss can improve synthetic image quality.

Regarding other artifacts generated during image translation, we noticed that all the considered models tried to modify the tools shape, adding or erasing parts from the simulation tools. Sample images of this phenomenon are presented in Fig. 4 and in the additional materials. Additive artifacts can be easily removed during the tool blending process, that copies only the parts of the tool covered by simulation masks, while erasing ones, that were present only in MUNIT* and MUNIT*+*SSIM*, do not have a fast and effective fix. This strengthened our decision to select cycle-GAN+*SSIM* models to produce synthetic datasets for ***E2***.

B. *E2* discussion

The results in ***E2*** showed the potential of the proposed methodology. Using only synthetic frames for both training and validation, we were able to obtain competitive results in all the considered datasets compared to the models trained on real data. In MICCAI 2017 the two models showed a ΔIoU score around 4% that can be partially explained by the presence of suturing needles in the scene, as shown in Fig. 6. Needles are indeed constantly present in the real frames (training/test), allowing the model to learn to not consider them as tools even though they have similar colors. The same did not happen however when the model was trained on synthetic frames, where needles were not encountered. However, the achieved 92% *IoU* score shows that the proposed dataset allowed the segmentation model to well generalize on challenging in-vivo frames with occlusions and tool-tissue interaction.

In MICCAI 2015, the lack in performances (4-5% ΔIoU) is mainly caused by the consistent mislabelling of part of the tools' shaft, as shown in the first column of Fig. 6. However, the training on synthetic frames still allowed the model to achieve appealing results compared to its competitor. This is shown again in Fig. 6, third column, where both the models were able to segment the EndoWrist® Monopolar Curved Scissors present in the FOV, that, in both cases, were not seen during the training procedure.

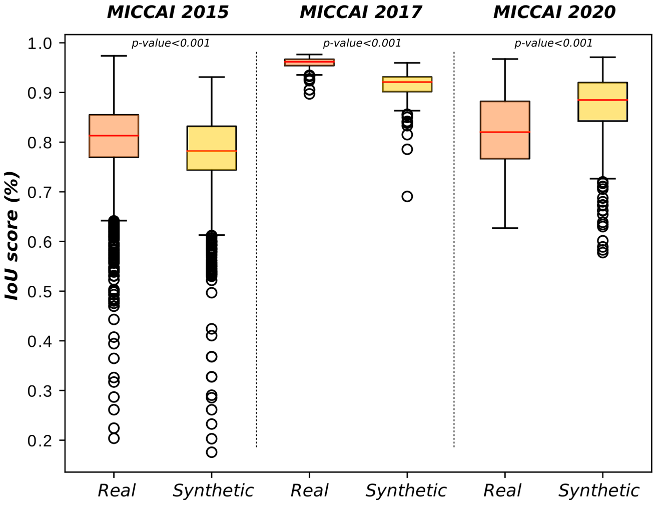


Fig. 7: Segmentation results on the considered datasets. Models' performances were evaluated in terms of Intersection over Union (*IoU*) score. All the results show statistically different distributions (one-way ANOVA test, p-value<0.001).

Finally, on MICCAI 2020 dataset, the model trained on synthetic frames outperformed the competitor by 6-7% ΔIoU, even though both the models were not able to correctly classify the parts covered by blood. However this was expected, since no blood was seen during the training phase in both cases. These results show that the proposed framework is robust even against human input variability and thus that realistic datasets can be produced even using recorded kinematic data as simulator input.

C. Limitations and future work

The main limitation for the proposed work is the need for segmentation ground truth in order to produce real domain sets suitable for I2I model training. Thus, advancing to a complete unsupervised framework would be a natural extension of this work. Another major constraint to our framework comes from the lack of a preferred and reliable evaluation metric for our

I2I models. A potential solution could be to evaluate them by training multiple segmentation models on the produced synthetic datasets, then selecting the best one based on the *IoU* score obtained on a real validation set. This however comes at the expense of further labelled data, that were not available for this work. Additionally, the framework could be extended to other surgical vision tasks such as 3D surgical tool pose estimation or action recognition. Additionally, temporal consistency, as described in [15], needs to be investigated further to move from single frame to full surgical video synthesis.

VII. Conclusions

In this paper, we propose a novel framework for synthetic surgical frames generation. Our methodology makes use of a robot virtual simulator along with I2I models to transfer the realistic style from ex-vivo and in-vivo images onto simulation tool frames. Once the simulated tools are passed through this transformation, they are blended on a surgical background to produce complete surgical images. Our method was validated by comparing the results obtained by two segmentation models trained on synthetic and real data respectively. We showed that the use of synthetic frames led the segmentation network to perform similarly to its competitor in challenging surgical scenarios.

Acknowledgment

The work was supported by the Wellcome/EPSRC Centre for Interventional and Surgical Sciences (WEISS) [203145Z/16/Z]; Engineering and Physical Sciences Research Council (EPSRC) [EP/P027938/1, EP/R004080 /1, EP/P012841/1]; The Royal Academy of Engineering Chair in Emerging Technologies Scheme; and Horizon 2020 FET (GA 863146). We thank Intuitive Surgical Inc and the dVRK community for their support of this work.

References

[1] D. Stoyanov, "Surgical vision," *Annals of biomedical engineering*, vol. 40, no. 2, pp. 332–345, 2012.

[2] L. Maier-Hein, S. S. Vedula, S. Speidel, N. Navab, R. Kikinis, A. Park, M. Eisenmann, H. Feussner, G. Forestier, S. Giannarou *et al.*, "Surgical data science for next-generation interventions," *Nature Biomedical Engineering*, vol. 1, no. 9, pp. 691–696, 2017.

[3] L. C. Garcia-Peraza-Herrera, W. Li, L. Fidon, C. Gruijthuijsen, A. Devreker, G. Attilakos, J. Deprest, E. Vander Poorten, D. Stoyanov, T. Vercauteren *et al.*, "Toolnet: holistically-nested real-time segmentation of robotic surgical tools," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5717–5722.

[4] M. Pfeiffer, I. Funke, M. R. Robu, S. Bodenstedt, L. Strenger, S. Engelhardt, T. Roß, M. J. Clarkson, K. Gurusamy, B. R. Davidson *et al.*, "Generating large labeled data sets for laparoscopic image processing tasks using unpaired image-to-image translation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 119–127.

[5] K. Lee, M.-K. Choi, and H. Jung, "Davincigan: Unpaired surgical instrument translation for data augmentation," in *International Conference on Medical Imaging with Deep Learning*, 2019, pp. 326–336.

[6] E. Colleoni, P. Edwards, and D. Stoyanov, "Synthetic and real inputs for tool segmentation in robotic surgery," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2020, pp. 700–710.

[7] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.

[8] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, "Neural style transfer: A review," *IEEE transactions on visualization and computer graphics*, 2019.

[9] I. Luengo, E. Flouty, P. Giataganas, P. Wisanuvej, J. Nehme, and D. Stoyanov, "Surreal: Enhancing surgical simulation realism using style transfer," in *British Machine Vision Conference 2018, BMVC 2018*. BMVA, 2018, pp. 1–12.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[11] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in neural information processing systems*, 2016, pp. 2234–2242.

[12] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[13] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 172–189.

[14] S. Lin, F. Qin, Y. Li, R. A. Bly, K. S. Moe, and B. Hannaford, "Lc-gan: Image-to-image translation based on generative adversarial network for endoscopic images," *arXiv preprint arXiv:2003.04949*, 2020.

[15] S. Engelhardt, R. De Simone, P. M. Full, M. Karck, and I. Wolf, "Improving surgical training phantoms by hyperrealism: deep unpaired image-to-image translation from real surgeries," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 747–755.

[16] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[17] T. Ozawa, Y. Hayashi, H. Oda, M. Oda, T. Kitasaka, N. Takeshita, M. Ito, and K. Mori, "Synthetic laparoscopic video generation for machine learning-based surgical instrument segmentation from real laparoscopic video and virtual surgical instruments," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, pp. 1–8, 2020.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[20] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[21] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[22] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.

[23] G. A. Fontanelli, M. Selvaggio, M. Ferro, F. Ficuciello, M. Vendittelli, and B. Siciliano, "A v-rep simulator for the da vinci research kit robotic platform," in *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*. IEEE, 2018, pp. 1056–1061.

[24] J. Ho Lee, I. Choi, and M. H. Kim, "Laplacian patch-based image synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2727–2735.

[25] M. Tomei, M. Cornia, L. Baraldi, and R. Cucchiara, "Art2real: Unfolding the reality of artworks via semantically-aware image-to-image translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5849–5859.

[26] Y. Zhang, Y. Yin, R. Zimmermann, G. Wang, J. Varadarajan, and S.-K. Ng, "An enhanced gan model for automatic satellite-to-map image conversion," *IEEE Access*, vol. 8, pp. 176 704–176 716, 2020.

[27] C.-S. Chiang and C.-S. D. Shih, "Using synthesized data to train deep neural net with few data," in *Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, 2020, pp. 19–25.

[28] A. Borji, "Pros and cons of gan evaluation measures," *Computer Vision and Image Understanding*, vol. 179, pp. 41–65, 2019.

[29] M. M. Mukaka, "A guide to appropriate use of correlation coefficient in medical research," *Malawi medical journal*, vol. 24, no. 3, pp. 69–71, 2012.