FAST AND DATA-EFFICIENT IMAGE SEGMENTATION

by

Daniil Pakhomov

A dissertation submitted to Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland

August, 2022

Abstract

Abundance and affordability of cameras has enabled scalable and affordable collection of image data. This has led to many research opportunities both in robot-assisted surgery and general computer vision domain related to image segmentation. In this thesis, we focus on image segmentation problem as it is a fundamental task which has many applications including pose estimation of surgical tools in robotic surgery and eye tracking in head mounted displays. As a result of our work we present a data-efficient method that does not require human annotation of data and exhibits real-time inference.

First, we introduce the use of residual neural networks for surgical instrument segmentation for robotic surgery. We show state of the art results on multiple instrument segmentation datasets. Second, we introduce a neural architecture search method that is able to find a very efficient image segmentation model capable of real-time inference. Real-time inference is a crucial requirement for image segmentation methods for robotic surgery. Third, to reduce the amount of annotation required for our method, we introduce a semi-supervised approach which leverages unlabeled images and synthetic training data. Finally, we introduce the use of generative adversarial networks for unsupervised discovery of segmentation classes from unlabeled image data. Here, we show for this first time that this task is possible without any annotated data. Data annotation for image segmentation is a very time consuming procedure as it requires every pixel of an image to be classified into one of the classes. We study the ability of recently introduced multimodal approaches like CLIP to assign text labels to our discovered segmentation regions. At the end, we present a model that is able to not only discover segmentation regions automatically but also assigns text labels them.

Primary Advisor: Nassir Navab (Johns Hopkins University)

Reader: Kemar E. Green (Johns Hopkins University)

Acknowledgments

I would like to thank my advisor, Nassir Navab, for his continued support and guidance throughout my PhD journey. With his guidance, I was able to find a topic that I was passionate about which later on allowed me to meaningfully contribute to that area of computer vision.

I would also like to thank Philipp Koehn, who gave me the chance to assist teaching his courses on artificial intelligence and natural language processing. I believe that this experience allowed me to broaden my knowledge and gain valuable teaching experience.

I'd also like to thank all of my mentors with whom I worked during my internships at Intuitive Surgical and Google: Max Allan, Mahdi Azizian, Yury Kartynnik, Karthik Raveendran and all of the Google MediaPipe team.

I'm also grateful to everyone that I had the chance to collaborate with during my time at Hopkins. I would especially like to thank Kemar E. Green, Sanchit Hira, Vittal Premachandran and Wei Shen.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter we formulate motivation for the thesis and provide a general outline, motivate the task of image segmentation for robotic surgery and other fields and show its importance. We highlight the questions that are raised and addressed by the thesis. We conclude with a concise thesis statement that emphasizes our main findings.

1.1 Motivation

Semantic segmentation plays crucial part in computer assisted interventions (CAI) during Robot-assisted Minimally Invasive Surgery (RMIS) as methods for full surgical instrument pose estimation rely on high quality segmentation of instrument parts [12]. Also, during CAI additional visual information is integrated like overlaying pre- and intra-operative imaging with surgical console which can improve decision making during complex procedures [13]. Segmentation of instruments also plays important part here as it is used to prevent rendered overlays from occluding the instruments [14]. At the same time, highly accurate eye segmentation is used as a part of gaze estimation in VR and AR devices [4]. Accurate gaze estimation allows to significantly reduce power consumption in AR and VR devices by only generating high quality picture at the region where a user is currently looking [4].

Segmentation of instruments and eye segmentation are extremely difficult as environment and patient can vary a lot and exhibit lighting changes or occlusions. All previous recent work relied on deep learning approaches to solve this problem [4,5,14]. Additional challenge related to all image segmentation problems is a dataset creation: it takes a lot of effort to create high quality annotations for images [15] [1].

In this thesis, we focus primarily on a problem of image segmentation, which is the task of labeling every pixel in the image with the class of its enclosing object or region. Image segmentation serves as a foundation for many important applications related to robotic surgery and augmented and virtual reality. Moreover, many applications lack annotated image segmentation training data and require real-time inference. Therefore, we aim to solve the problem of image segmentation under no or limited annotated training data availability and real-time inference constraints.

1.2 Outline

In this section we pose the main questions that are addressed in the thesis. We highlight the chapters of the thesis that are relevant to each raised question.

1.2.1 Is it possible to accurately perform a fully automatic and accurate segmentation of surgical instruments?

Automatic segmentation of surgical instruments is a crucial step towards full instrument pose estimation but it can also be solely used to improve user interactions with the robotic system. Leveraging techniques such as deep residual learning and dilated convolutions to advance both binary-segmentation and instrument part segmentation performance is the subject of Chapter 2.

1.2.2 Can we automatically learn an efficient real-time network architecture for instrument segmentation?

While previous research focused primarily on methods that deliver high accuracy segmentation masks, majority of them can not be used for real-time applications due to their computational cost. Should we manually design a network architecture for each task or is it possible to learn an optimal architecture tailored for a given task? This question is addressed in Chapter 3.

1.2.3 Is instrument segmentation possible with inaccurate synthetic annotations?

While many recent approaches based on convolutional neural networks have shown great results, a key barrier to progress lies in the acquisition of a large number of manually-annotated images which is necessary for an algorithm to generalize and work well in diverse surgical scenarios. Unlike the surgical image data itself, annotations are difficult to acquire and may be of variable quality. On the other hand, synthetic annotations can be automatically generated by using forward kinematic model of the robot and CAD models of tools by projecting them onto an image plane. Unfortunately, this model is very inaccurate and cannot be directly used for supervised learning of image segmentation models. Is it possible to design an algorithm that will learn accurate segmentation model even from corrupted data? And if so, how will the accuracy differ from fully supervised methods? These questions are addressed in Chapter 4.

1.2.4 Is image segmentation possible even in the absence of annotations?

While fully supervised methods show impressive results, their applicability is limited by a very slow process of image annotation. This issue is more pronounced in the medical imaging community where a domain knowledge is required to perform annotations. Can we automatically segment images into semantically meaningful regions without human supervision? This question is addressed in Chapter 5.

1.3 Thesis Statement

Image segmentation can be performed in an automated fashion under constraints of limited or no training data availability and real-time inference.

Chapter 2

Deep Residual Learning for Instrument Segmentation in Robotic Surgery

In this chapter, we focus on the task of labeling every pixel in the image with the class of its enclosing object or region, which is also known as image segmentation in the literature. In this particular case, we focus on binary instrument segmentation, where the objective is to label every pixel as instrument or background and instrument part segmentation, where different semantically separate parts of the instrument are labeled. We introduce dilated deep residual neural networks for this task and show that they are capable of achieving state-of-the-art performance.

2.1 Introduction

Robot-assisted Minimally Invasive Surgery (RMIS) overcomes many of the limitations of traditional laparoscopic Minimally Invasive Surgery (MIS), providing the surgeon with improved control over the anatomy with articulated instruments and dexterous master manipulators. In addition to this, 3D-HD visualization on systems such as da Vinci enhances the surgeon's depth perception and operating precision [16]. However, complications due to the reduced field-of-view provided by the surgical camera limit the surgeon's ability to self-localize. Traditional haptic cues on tissue composition are lost through the robotic control system [17].

Overlaying pre- and intra-operative imaging with the surgical console can provide the surgeon with valuable information which can improve decision making during complex procedures [13]. However, integrating this data is a complex task and involves understanding spatial relationships between the surgical camera, operating instruments and patient anatomy. A critical component of this process is segmentation of the instruments in the camera images which can be used to prevent rendered overlays from occluding the instruments while providing crucial input to instrument tracking frameworks [12, 18].

Segmentation of surgical tools from tissue backgrounds is an extremely difficult task due to lighting challenges such as shadows and specular reflections, visual occlusions such as smoke and blood (see Fig. 2.1). Early methods attempted to simplify the problem by modifying the appearance of the instruments [19]. However, this

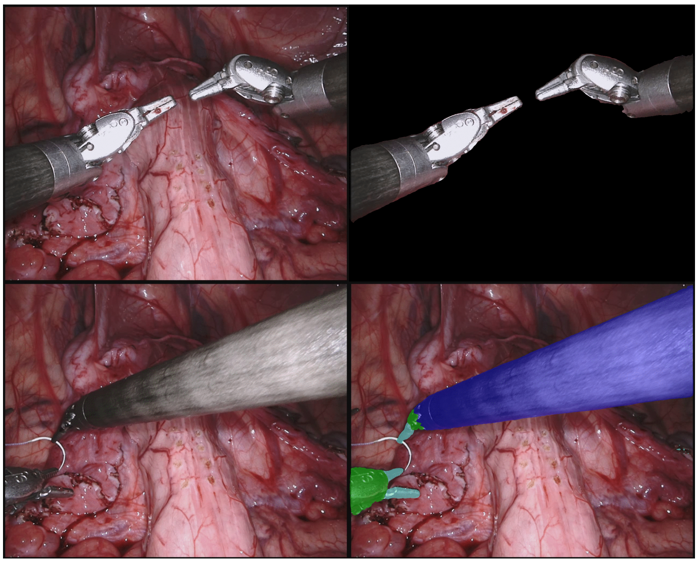


Figure 2.1: Example frames from RMIS procedures present in the dataset. Left column shows example images frames from the dataset. Right column shows the binary and instrument part segmentation of corresponding images delivered by our method.

complicates clinical application of the technique as sterilization can become an issue. Segmentation of the instruments using natural appearance is a more desirable approach as it can be applied directly to pre-existing clinical setups. However, this defines a more challenging problem. To solve it, previous work has relied on machine learning techniques to model the complex discriminative boundary. The instrument-background segmentation can be modeled as a binary segmentation problem to which discriminative models, such as Random Forests [20], maximum likelihood Gaussian Mixture Models [18] and Naive Bayesian classifiers [21], all trained on color features, have been applied. More recently, the state-of-the-art has increasingly been defined by Fully Convolutional Networks (FCNs), such as the FCN-8s model [22] adapted for the task of binary segmentation of robotic tools [23] and U-Net [24] which was used for both binary and instrument part segmentation [25].

In this work, we adopt the state-of-art residual image classification Convolutional Neural Network (CNN) [26] for the task of semantic image segmentation by casting it into a FCN. However, the transformed model delivers a prediction map of significantly reduced dimension compared to the input image [22]. To account for that, we reduce in-network downsampling, employ dilated (atrous) convolutions to enable initialization with the parameters of the original classification network, and perform simple bilinear interpolation of the feature maps to obtain the original image size [27, 28]. This approach is a powerful alternative to using deconvolutional layers (upsampling layers) and "skip connections" as in FCN-8s model [22] and CSL model [25]. By employing it, we advance the state-of-the-art in binary and instrument part segmentation of tools on the EndoVis 2017 Robotic Instruments dataset [1].

2.2 Method

The goal of this work is to label every pixel of an image $\mathbf{I}$ with one of C semantic classes, representing surgical tool part or background. In case of binary segmentation, the goal is to label each pixel into $C = 2$ classes, namely surgical tool and background. In this work, we also consider a more challenging multi-class segmentation with $C = 4$ classes, namely tool's shaft, wrist and jaws and background.

Each image $\mathbf{I}_i$ is a three-dimensional array of size $h \times w \times d$, where h and w are spatial dimensions, and d is a channel dimension. In our case, $d = 3$ because we use RGB images. Each image $\mathbf{I}_i$ in the training dataset has corresponding annotation $\mathbf{A}_i$ of a size $h \times w \times C$ where each element represents one-hot encoded semantic label $a \in \{0, 1\}^C$ (for example, if we have classes 1, 2, and 3, then the one-hot encoding of label 2 is $(0, 1, 0)^T$).

We aim at learning a mapping from $\mathbf{I}$ to $\mathbf{A}$ in a supervised fashion that generalizes to previously unseen images. In this work, we use CNNs to learn a discriminative classifier which delivers pixel-wise predictions given an input image. Our method is built upon state-of-the-art deep residual image classification CNN (ResNet-18, Section 2.2.1), which we convert into fully convolutional network (FCN, Section 2.2.2).

CNNs reduce the spatial resolution of the feature maps by using pooling layers or convolutional layers with strides greater than one. However, for our task of pixel-wise prediction we would like dense feature maps. We set the stride to one in the last two layers responsible for downsampling, and in order to reuse the weights from a pre-trained model, we dilate the subsequent convolutions (Sec. 3.2.1) with an appropriate rate. This enables us to obtain predictions that are downsampled only by a factor of 8× (in comparison to the original downsampling of 32×).

We then apply bilinear interpolation to regain the original spatial resolution. With an output map of the same resolution as an input image, we perform end-to-end training by minimizing the normalized pixel-wise cross-entropy loss [22].

2.2.1 Deep Residual Learning

Traditional CNNs learn filters that process the input x_l and produce a filtered response x_{l+1}, as shown below

$$y_l = g(x_l, w_l), \tag{2.1}$$

$$x_{l+1} = f(y_l). \tag{2.2}$$

Here, $g(.,.)$ is a standard convolutional layer with w_l being the weights of the layer's convolutional filters and biases, $f(.)$ is a non-linear mapping function such as the Rectified Linear Unit (ReLU). Recently modifications [26] showed that significant gains in performance can be obtained by employing "residual units" as a building block of a deep CNN, and called such networks Residual Networks (ResNets). Each unit of a ResNet can be expressed in the following general form

$$y_l = h(x_l) + F(x_l, W_l), \tag{2.3}$$

$$x_{l+1} = f(y_l), \tag{2.4}$$

where x_l and x_{l+1} are input and output of the l-th unit, and $F(.,)$ is a residual function to be learnt. The function $h(.)$ is a simple identity mapping, $h(x_l) = x_l$ and $f(.)$ is a rectified linear unit activation (ReLU) function. As $h(x_l)$ is chosen to be an identity mapping, it is easily realized by attaching an identity skip connection (also known as a “shortcut” connection). In our work, we adopt ResNet-18 architecture which allows us to achieve state-of-the-art performance in tool segmentation.

2.2.2 Fully Convolutional Networks

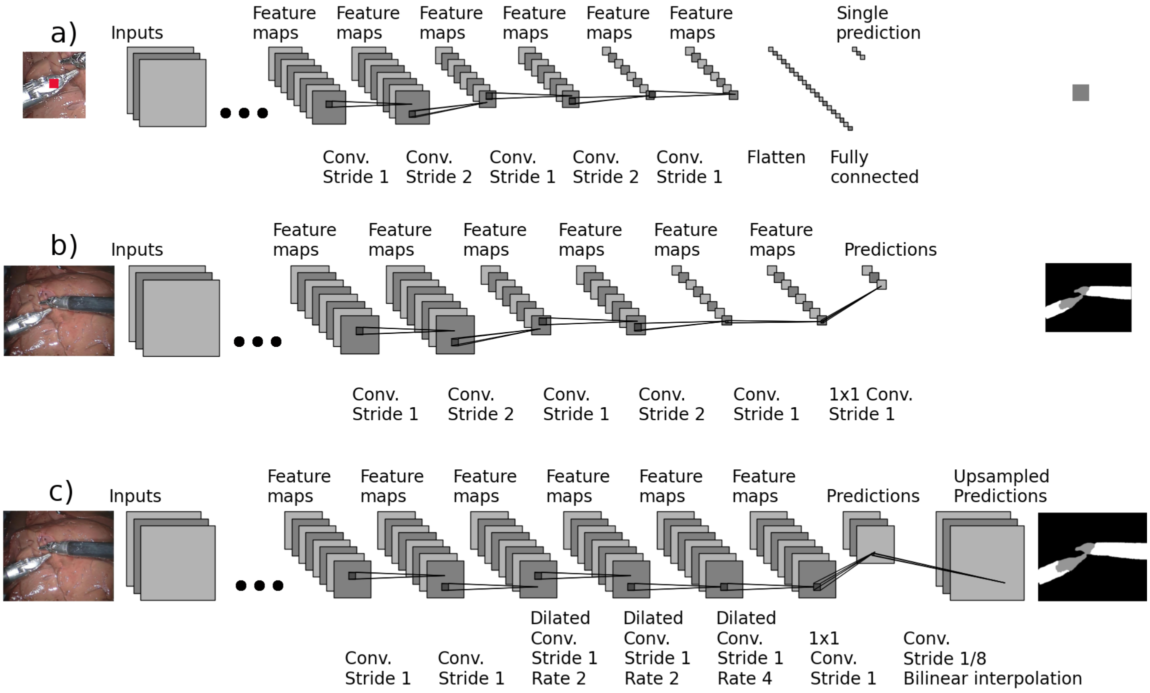


Figure 2.2: A simplified CNN before and after being converted into an FCN (illustrations **(a)** and **(b)** respectively), after reducing downsampling rate with integration of dilated convolutions into its architecture with subsequent bilinear interpolation (illustration **(c)**). Illustration **(a)** shows an example of applying a CNN to an image patch centered at the red pixel which gives a single vector of predicted class scores. Illustration **(b)** shows the fully connected layer being converted into 1×1 convolutional layer, making the network fully convolutional, thus enabling a dense prediction. Illustration **(c)** shows network with reduced downsampling and dilated convolutions that produces outputs that are being upsampled to acquire pixelwise predictions.

Deep CNNs (e.g. AlexNet, VGG16, ResNets, etc.) are primarily trained for the task of image classification. However, to obtain the output granularity required for a task such as image segmentation requires converting the CNN’s fully connected layers into convolutions with kernels that are equal to their fixed input regions [22] which creates an FCN. An FCN operates on inputs of any size, and produces an output with reduced spatial dimensions [22].

Fully convolutional models deliver prediction maps with significantly reduced dimensions (for both VGG16 and ResNets, the spatial dimensions are reduced by a factor of 32). In the previous work [22], it was shown that adding a deconvolutional layer to learn the upsampling with factor 32 provides a way to get the prediction map of original image dimension, but the segmentation boundaries delivered by this approach are usually too coarse. To tackle this problem, two approaches were recently developed which are based on modifying the architecture. (i) By fusing features from layers of different resolution to make the predictions [22, 25]. (ii) By avoiding downsampling of some of the feature maps [27, 28] (removing certain pooling layers in VGG16 and by setting the strides to one in certain convolutional layers responsible for the downsampling in ResNets). However, since the weights in the subsequent layers were trained to work on a downsampled feature map, they need to be adapted to work on the feature maps of a higher spatial resolution. To this end, [27] employs dilated convolutions. In our work, we follow the second approach: we mitigate the decrease in the spatial resolution by using convolutions with strides equal to one in the last two convolutional layers responsible for downsampling in ResNet-18 and by employing dilated convolutions for subsequent convolutional layers (Sec. 3.2.1).

2.2.3 Dilated Convolutions

In order to account for the problem stated in the previous section, we use dilated (atrous) convolution. Dilated convolution[1] in one-dimensional case is defined as

$$y[i] = \sum_{k=1}^{K} x[i + rk]w[k]$$

where, x is an input 1D signal, y output signal and w is a filter of size K. The rate parameter r corresponds to the dilation factor. The dilated convolution operator can reuse the weights from the filters that were trained on downsampled feature maps by sampling the unreduced feature maps with an appropriate rate.

In our work, since we choose not to downsample in some convolutional layers (by setting their stride to one instead of two), convolutions in all subsequent layers are dilated. This enables initialization with the parameters of the original classification network, while producing higher-resolution outputs. This transformation follows [27] and is illustrated in Fig. 2.2c.

2.2.4 Training

Given a sequence of images $\{\mathbf{I}_t\}_{t=0}^{n_t}$, and sequence of ground-truth segmentation annotations $\{\mathbf{A}_t\}_{t=0}^{n_t}$, we optimize normalized pixel-wise cross-entropy loss [22] using Adam optimization algorithm [29] with learning rate set to 10^{-4} (n_t stands for the

[1] We follow the practice of previous work and use simplified definition without mirroring and centering the filter [27].

number of training examples). Other parameters of Adam optimization algorithm were set to the values suggested in [29].

2.3 Experiments and Results

We test our method on the EndoVis 2017 Robotic Instruments dataset [1]. The training dataset consists of 8 high resolution (1280×1024) sequences with 225 frames each that were acquired from a da Vinci Xi surgical system during several different procedures [1]. Each pixel is labeled as either tool's shaft, wrist and jaws or background. The test dataset consists of 8 75-frame sequences sampled immediately after each training sequence and 2 full 300-frame sequences. Following the terms of the challenge, we excluded the corresponding training set when evaluating on one of the 75-frame sequences.

2.3.1 Results

We report our results for binary and instrument part segmentation in Tab. 4.1 and Tab. 2.2 respectively using standard metric such as Intersection Over Union (IoU). We can see that our method outperforms previous work. Fig. 2.1 shows some qualitative results for both the binary segmentation and the instrument part segmentation tasks.

	NCT	UB	BIT	MIT	SIAT	UCL	TUM	Delhi	UA	UW	Ours
Dataset 1	0.784	0.807	0.275	0.854	0.625	0.631	0.760	0.408	0.413	0.337	**0.862**
Dataset 2	0.788	0.806	0.282	0.794	0.669	0.645	0.799	0.524	0.463	0.289	**0.855**
Dataset 3	0.926	0.914	0.455	**0.949**	0.897	0.895	0.916	0.743	0.703	0.483	0.926
Dataset 4	0.934	0.925	0.310	**0.949**	0.907	0.883	0.915	0.782	0.751	0.678	0.931
Dataset 5	0.701	0.740	0.220	0.862	0.604	0.719	0.810	0.528	0.375	0.219	**0.877**
Dataset 6	0.876	0.890	0.338	**0.922**	0.843	0.852	0.873	0.292	0.667	0.619	0.896
Dataset 7	0.846	**0.930**	0.404	0.856	0.832	0.710	0.844	0.593	0.362	0.325	0.869
Dataset 8	0.881	0.904	0.366	0.937	0.513	0.517	0.895	0.562	0.797	0.506	**0.939**
Dataset 9	0.789	0.855	0.236	0.865	0.839	0.808	0.877	0.626	0.539	0.377	**0.879**
Dataset 10	0.899	**0.917**	0.403	0.905	0.899	0.869	0.909	0.715	0.689	0.603	0.915
Mean IOU	0.843	0.875	0.326	0.888	0.803	0.785	0.873	0.612	0.591	0.461	**0.896**

Table 2.1: Quantitative results of our method and comparison with previous state-of-the-art in binary segmentation of robotic tools [1].

	NCT	UB	BIT	MIT	SIAT	UCL	TUM	UA	UW	Ours
Dataset 1	0.723	0.715	0.317	0.737	0.591	0.611	0.708	0.485	0.235	**0.791**
Dataset 2	0.705	0.725	0.294	**0.792**	0.632	0.606	0.740	0.559	0.244	0.785
Dataset 3	0.809	0.779	0.319	**0.825**	0.753	0.692	0.787	0.640	0.239	0.805
Dataset 4	0.845	0.737	0.304	0.902	0.792	0.630	0.815	0.692	0.238	**0.920**
Dataset 5	0.607	0.565	0.280	0.695	0.509	0.541	0.624	0.473	0.240	**0.734**
Dataset 6	0.731	0.763	0.271	0.802	0.677	0.668	0.756	0.608	0.235	**0.832**
Dataset 7	0.729	**0.747**	0.359	0.655	0.604	0.523	0.727	0.438	0.207	0.641
Dataset 8	0.644	0.721	0.300	0.737	0.496	0.441	0.680	0.604	0.236	**0.855**
Dataset 9	0.561	0.597	0.273	0.650	0.655	0.600	**0.736**	0.551	0.221	0.660
Dataset 10	0.788	0.767	0.273	0.762	0.751	0.713	**0.807**	0.637	0.241	0.806
Mean IOU	0.699	0.700	0.289	0.737	0.667	0.623	0.751	0.578	0.357	**0.764**

Table 2.2: Quantitative results of our method and comparison with previous state-of-the-art in the parts based segmentation of robotic tools [1].

2.4 Discussion and Conclusion

In this work, we propose a method to perform robotic tool segmentation. This is an important task, as it can be used to prevent rendered overlays from occluding the instruments or to estimate the pose of a tool [12]. We use deep network to model the mapping from the raw images to the segmentation maps. Our use of a state-of-the-art deep network (ResNet-18) with dilated convolutions helps us achieve improvement in binary tool and instrument part segmentation over the previous stat-of-the-art. Our results show the benefit of using deep residual networks for this task and also provide a solid baseline for the future work.

Chapter 3

Searching for Efficient Architecture for Instrument Segmentation in Robotic Surgery

In the previous chapter we explored dilated residual neural networks for the task of instrument segmentation. An important question is whether this task can be performed in real-time since many applications related to surgical robotics have this as a requirement. In this section, we analyze the use of hyperparameters that are usually empirically selected by researchers when they design a neural network. Instead, we introduce a method that is capable of learning these hyperparameters automatically, allowing to find an optimal architecture for real-time instrument segmentation. Our discovered neural network architectures exhibit superior performance compared to

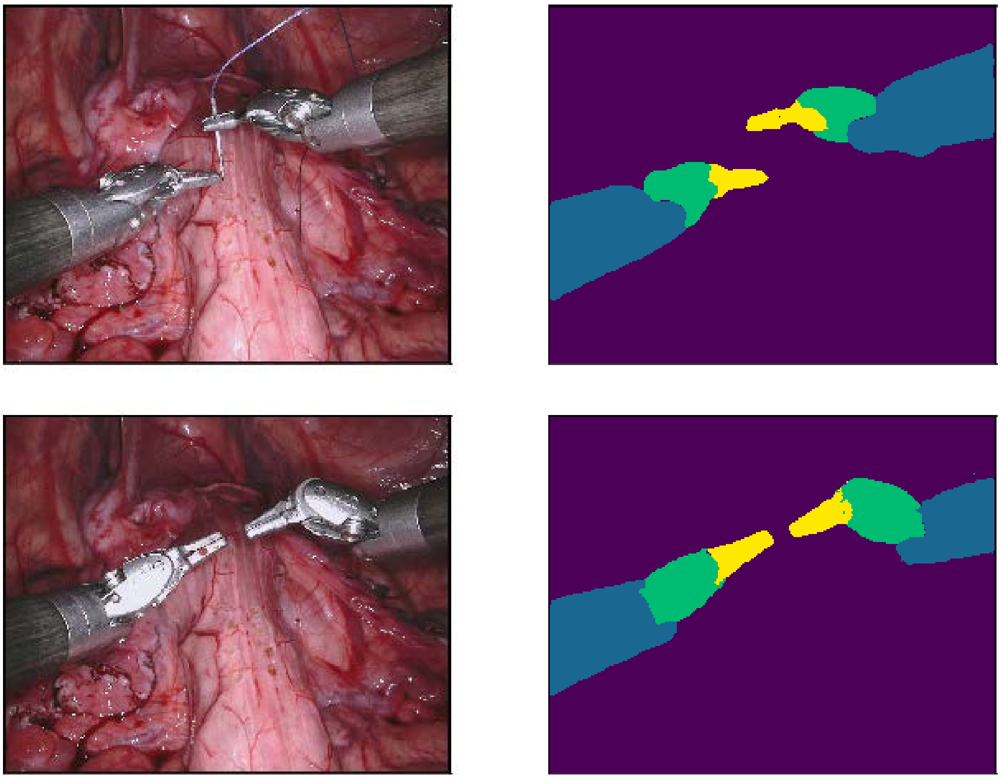


Figure 3.1: Multi-class instrument segmentation results delivered by our method on sequences from validational subset of Endovis 2017 dataset.

previously introduced methods in terms of speed and accuracy trade off.

3.1 Introduction

Robot-assisted Minimally Invasive Surgery (RMIS) provides a surgeon with improved control, facilitating procedures in confined and difficult to access anatomical regions. However, complications due to the reduced field-of-view provided by the surgical camera limit the surgeon's ability to self-localize. Computer assisted interventions (CAI) can help a surgeon by integrating additional information. For example,

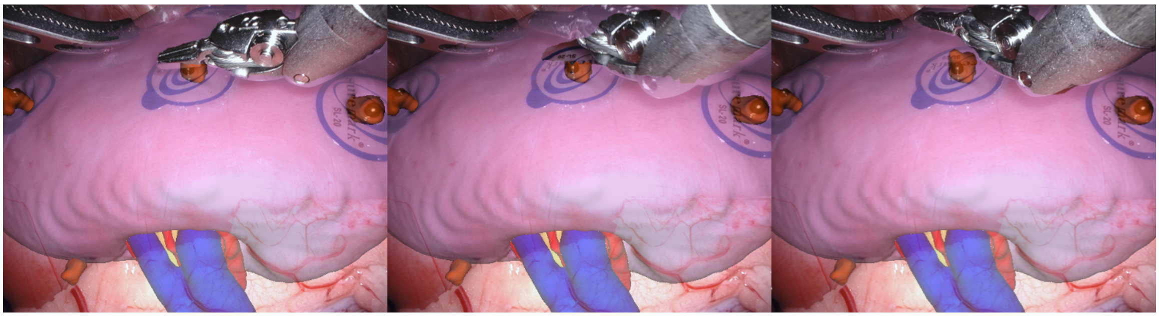


Figure 3.2: An example image sequence showing previous state-of-the-art algorithm masking an overlay of a porcine kidney. As the instrument tip is quickly rotated, the mask lags behind the video as the video frame rate of 60 Hz is faster than the network can handle, resulting in dropped frames.

overlaying pre- and intra-operative imaging with the surgical console can provide a surgeon with valuable information which can improve decision making during complex procedures [13]. Integrating this data is a complex task and involves understanding relations between the patient anatomy, operating instruments and surgical camera. Segmentation of the instruments in the camera images is a crucial component of this process and can be used to prevent rendered overlays from occluding the instruments while providing crucial input to instrument tracking frameworks [12, 18].

There has been a significant development in the field of instrument segmentation [5, 25, 30] based on recent advancement in deep learning [26, 27]. Additionally, release of new datasets for surgical tools segmentation with high-resolution challenging images further improved the field by allowing methods to be more rigorously tested [1]. While the state-of-the-art methods show impressive pixel accurate results [5, 30], the inference time of these methods makes them unsuitable for real-time applications. In our work we address this problem and present a method that has a faster than real time inference time while delivering high quality segmentation masks.

3.2 Method

In this work, we focus on the problem of surgical instrument segmentation. Given an input image, every pixel has to be classified into one of C mutually exclusive classes. We consider two separate tasks with increasing difficulty: binary tool segmentation and multi-class tool segmentation. In the first case, each pixel has to be classified into $C = 2$ classes as belonging to surgical background or instrument. In the second task, into $C = 4$ classes, namely tool's shaft, wrist, jaws and surgical background. Our method is designed to perform these tasks efficiently while delivering high quality results. First, we discuss the previous state-of-the-art method based on dilated residual networks and highlight the major factor that makes it computationally expensive. Then, we present light residual networks that allow to solve this problem and make the method faster. To account for reduced accuracy we introduce a search for optimal dilation rates in our model which allows to improve its accuracy.

3.2.1 Dilated Residual Networks

We improve upon previous state-of-the-art approach [5] based on dilated residual network that employs ResNet-18 (see Fig. 3.3), a deep residual network pretrained on ImageNet dataset. The network is composed of four successive residual blocks, each one consisting of two residual units of ”basic” type [26]. The average pooling layer is removed and stride is set to one in the last two residual blocks responsible for downsampling, subsequent convolutional layers are dilated with an appropriate rate as suggested in [5, 27]. This allows to obtain predictions that are downsampled only by a factor of 8× (in comparison to the original downsampling of 32×) which makes the network work on a higher resolution features maps (see Fig. 3.3) and deliver finer predictions [5, 27].

The aforementioned method delivers very accurate segmentation masks but is too computationally expensive for real-time applications. The main reason for this is that it uses deep residual classification models pretrained on ImageNet that usually have great number of filters at the last layers: when the model is being transformated into dilated residual network, the downsampling operations at the last two residual blocks are removed forcing the convolutional layers to operate on the input that is spatially bigger by a factor of two and four respectively [5, 27] (see Fig. 3.3). Even the most shallow deep residual network ResNet-18 is too computationally expensive when converted into dilated residual network. This motivates us to create a smaller deep residual network that can be pretrained on the imageNet and converted into dilated residual model that exhibits improved running time.

3.2.2 Light Residual Networks

In order to solve the aforementioned problem, we introduce a light residual network, a deep residual model which satisfies the requirements:

- **Low latency:** exhibits low latency inference on high-resolution images when converted into dilated residual network. This is achieved by reducing the number of filters in the last stages of the network (see Fig. 3.3). In the original resnet networks [26], the number of filters in the last stages is considerable: after being converted into dilated versions, these stages experience increase in the computational price by a factor of two and four [27]. Since last stages are the biggest factor responsible for the increased inference time, we decrease the number of channels in them. This significantly decreases the inference time. While we noticed a considerable decrease in the accuracy of the model on the ImageNet dataset, the performance on the segmentation dataset only moderately decreased. We attribute it to the fact the number of filters in the last layers is of significant importance for the imagenet classification task because it needs to differentiate between one thousand classes compared to only four in our case.

- **Low GPU memory consumption:** the memory requirements of the network allow it to be trained with optimal batch and image crop sizes on single GPU device. Dilated residual networks consume a considerable amount of memory when trained for segmentation task [27], while still requiring relatively big batch size and crop size [31]. Similar to the previous section, the biggest factor responsible for the memory comsumption again involves the last stages of the network [27]. Since every activation has to be stored in memory during backpropagation [32], the layers that generate the biggest activations are contributing the most to the increased memory consumption. The last layers of the residual network work on increased spatial resolution after being converted to the dilated residual network and have more channels than other layers. By decreasing the number of channels in the last layers, we solve the problem and are able to train the model with sufficient batch size and image crop size [31].

- **ImageNet pretraining:** the network is pretrained on ImageNet dataset which was shown to be essential for good performance on small segmentation datasets like Endovis [30]. We pretrain all our models on the ImageNet dataset following the parameters suggested in [26].

We present two versions of Light ResNet-18 named Light ResNet-18-v1 and Light ResNet-18-v2 with number of channels in the last layers set to 64 and 32 respectively. Second version exhibits improved runtime speed at the expense of decreased accuracy on the segmentation task. All models were pretrained on imageNet dataset and then converted into dilated residual networks following [5]. After being converted, the network were trained on the Endovis 2017 [1] segmentation dataset for binary and multi-class instrument segmentation tasks. While being fast, the networks exhibit

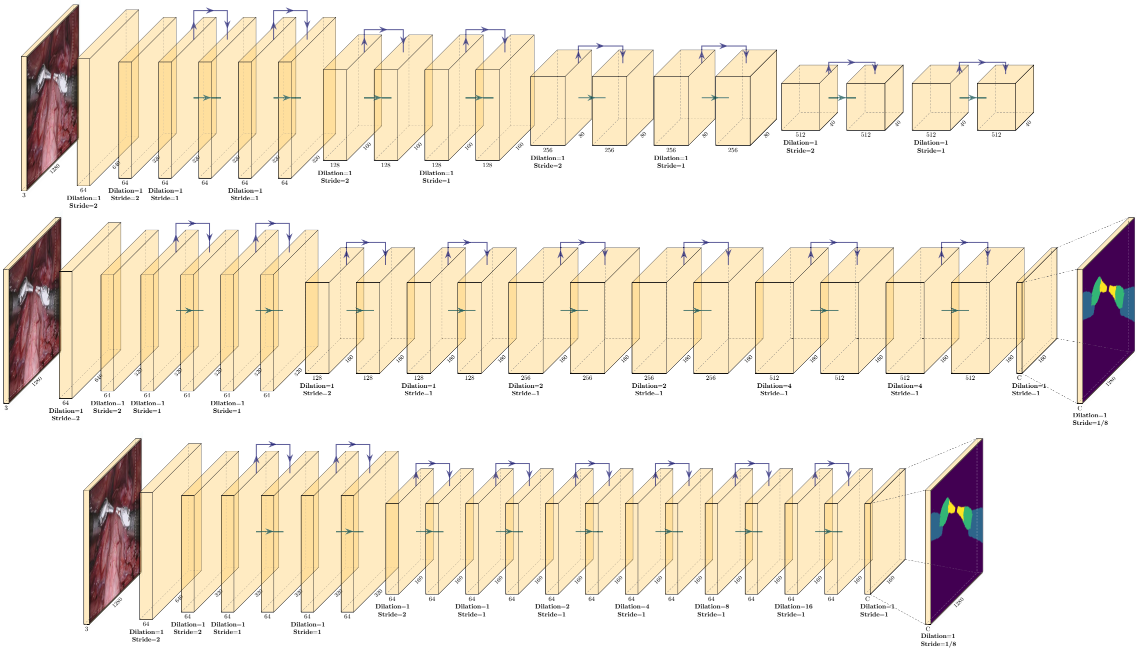


Figure 3.3: **Top row:** simplified ResNet-18 before being converted into dilated fully convolutional network (FCN). **Middle row:** ResNet-18 after being converted into FCN following the method of [5]. Since the stride was set to one in two layers and the number of channels was left the same, convolutional layers are forced to work on a greater resolution, therefore, computational overhead is substantially increased, making the network unable to deliver real-time performance. **Bottom row:** Our light-weight ResNet-18 being converted into dilated FCN and dilation rates are set to the values found during our differentiable search. The decreased number of channels makes the network fast while the found dilation rates increase its accuracy without any additional parameters or computational overhead. Green arrows represent the residual unit of a "basic" type, black arrows represent skip connections. First two layers does not have any residual units and are simplified in the figure. Dashed lines represent the bilinear upsampling operation. The figure is better viewed in a pdf viewer in color and zoomed in for details. The figure in bigger resolution is available in the supplementary material.

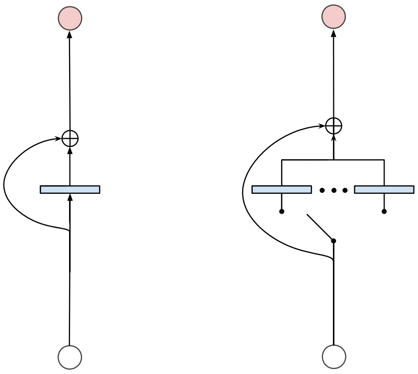


Figure 3.4: **Left:** Simplified residual unit, a building block of deep residual networks which are usually formed as a sequence of residual units. **Right:** A group of residual units with different dilation rates combined with a discrete decision gate which forces the network to choose only one of residual units depending on which of them leads to a better overall performance of the network. The gate is controlled by a variable which receives gradients during training. During optimization each layer has its own set of residual units and gate variables and by the end of training only the residual units with dilation rates that perform best are left (other choices can be safely removed). We perform search for dilation rates only for the last four residual units of our network.

reduced accuracy compared to state-of-the-art methods. In order to account for that and improve the accuracy without reducing the speed of the network, we search for optimal dilation rates for each layer.

3.2.3 Searching for Optimal Dilation Rates

In order to improve accuracy of our model further but avoid adding any new parameters or additional computational overhead [33], we search for optimal integer dilation rates for each residual unit. Since trying out all possible combinations of dilation rates and retraining a model each time is infeasible, we formulate the problem of dilation rate search as an optimization problem [34].

First, we update residual units of our model. Original residual unit [26] can be expressed in a form (see Figure 3.4):

$$x_{l+1} = x_l + \mathcal{F}(x_l) \tag{3.1}$$

where x_l and x_{l+1} are input and output of the l-th unit, and $\mathcal{F}$ is a residual function.

In order to allow our network to choose residual units with different dilation rates, we introduce gated residual unit which has N different residual connections with different dilation rates:

$$x_{l+1} = x_l + \sum_{i=0}^{N} \mathbf{Z}_i \cdot \mathcal{F}_i(x_l) \tag{3.2}$$

$$\sum_{i=0}^{N} \mathbf{Z}_i = 1 \tag{3.3}$$

$$\forall i \ \mathbf{Z}_i \in \{0, 1\} \tag{3.4}$$

where $\mathbf{Z}$ is a gate that decides which residual connection to choose and is represented with discrete one-hot-encoded categorical variable, N is equal to the number of dilation rates that we consider. In order to be able to search for the best dilation rates, we need to also optimize the gate variable $\mathbf{Z}$ which is not differentiable since it represents a hard decision. But it is possible to obtain estimated gradients of such hard decisions by introducing petrtubations in the system in the form of noise [34–37]. To be more specific, we use recent work that introduces differentiable approximate sampling mechanism for categorical variables based on a Gumbel-Softmax distribution (also known as a concrete distribution) [35–37]. We use Gumbel-Softmax to relax the discrete distribution to be continuous and differentiable with reparameterization trick:

$$\bar{\mathbf{Z}}_i = softmax((\log \alpha_i + G_i)/\tau) \tag{3.5}$$

Where G_i is an ith Gumbel random variable, $\bar{\mathbf{Z}}$ is the softened one-hot random variable which we use in place of $\mathbf{Z}$, α_i is a parameter that controls which residual unit to select and is optimized during training, τ is the temperature of the softmax, which is steadily annealed to be close to zero during training.

We update last four layers or our network to have gated residual units. Each of them allows the network to choose from a predefined set of dilation rates which we set to $\{1, 2, 4, 8, 16\}$. We train the network on the Endovis 2017 dataset by optimizing all weights including the α_i variables which control the selection of dilation rates. Upon convergence, the best dilation rates are decoded from the α_i variables. Discovered dilation rates can be seen at the Fig. 3.3. Next we train the residual network with specified dilation rates and original residual units.

3.2.4 Training

After the optimal dilation rates are discovered we update Light ResNet-18-v1 and ResNet-18-v2 to use them. Light ResNet-18-v1 and ResNet-18-v2 were first pre-trained on Imagenet. We train networks on the Endovis 2017 train dataset [1]. During training we recompute the batch normalization statistics [31,38]. We optimize normalized pixel-wise cross-entropy loss [27] using Adam optimization algorithm. Random patches are cropped from the images [27] for additional regularization. We employ crop size of 799. We use the 'poly' learning rate policy with an initial learning rate of 0.001 [31]. The batch size is set to 32.

3.3 Experiments and Results

We test our method on the EndoVis 2017 Robotic Instruments dataset [1]. There are 10 75-frame sequences in the test dataset that features 7 different robotic surgical instruments [1]. Samples from the dataset and qualitative results of our method are depicted in Fig. 4.1. We report quantitative results in terms of accuracy and inference time of our method.

As it can be seen our method is able to deliver pixel accurate segmentation while working at an extremely fast frame rate of up to 125 FPS.

Table 3.1: Quantitative results of our method compared to other approaches in terms of accuracy (measured using mean intersection over union metric) and latency (measured in miliseconds). Latency was measured for an image of input size 1024 × 1280 using NVIDIA GTX 1080Ti GPU. It can be seen that our light backbone allows for significantly decreased inference time, while learnt dilations help to improve decreased accuracy.

	Binary segmentation		Parts segmentation	
Model	IOU	Time	IOU	Time
TernausNet-16 [1,30]	0.888	184 ms	0.737	202 ms
Dilated ResNet-18 [5]	**0.896**	126 ms	**0.764**	126 ms
Dilated Light ResNet-18-v1	0.821	17.4 ms	0.728	17.4 ms
Light ResNet-18-v1 w/ Learnt Dilations	0.869	17.4 ms	0.742	17.4 ms
Dilated Light ResNet-18-v2	0.805	**11.8 ms**	0.706	**11.8 ms**
Light ResNet-18-v2 w/ Learnt Dilations	0.852	**11.8 ms**	0.729	**11.8 ms**

Table 3.2: Latency of our method as measured on a modern NVIDIA Tesla P100 GPU for an image of input size 1024 × 1280. We can see that fastest of our models is able to work at 125 frames per second.

	Binary segmentation		Parts segmentation	
Model	IOU	Time	IOU	Time
Light ResNet-18-v1 w/ Learnt Dilations	0.869	11.5 ms	0.742	11.5 ms
Light ResNet-18-v2 w/ Learnt Dilations	0.852	**7.95 ms**	0.729	**7.95 ms**

3.4 Discussion and Conclusion

In this work, we propose a method to perform real-time robotic tool segmentation on high resolution images. This is an important task, as it allows the segmentation results to be used for applications that require low latency, for example, preventing rendered overlays from occluding the instruments or estimating the pose of a tool [12]. We introduce a lightweight deep residual network to model the mapping from the raw images to the segmentation maps that is able to work at high frame rate. Additionally, we introduce a method to search for optimal dilation rates for our lightweight model, which improves its accuracy in binary tool and instrument part segmentation. Our results show the benefit of our method for this task and also provide a solid baseline for the future work.

Chapter 4

Towards Unsupervised Learning for Instrument Segmentation in Robotic Surgery with Cycle-Consistent Adversarial Networks

In the previous chapter we were able to automatically learn an optimal architecture for the real-time instrument segmentation. While the proposed method have shown great results, a key barrier to progress lies in the acquisition of a large number of manually-annotated images which is necessary for an algorithm to generalize and work well in diverse surgical scenarios. Unlike the surgical image data itself, annotations are difficult to acquire and may be of variable quality. On the other hand, synthetic annotations can be automatically generated by using forward kinematic model of the robot and CAD models of tools by projecting them onto an image plane. Unfortunately, this model is very inaccurate and cannot be directly used for supervised learning of image segmentation models. In this chapter we introduce an algorithm that is able to learn accurate segmentation model even from corrupted data.

4.1 Introduction

Robot-assisted Minimally Invasive Surgery (RMIS) provides a surgeon with improved control, facilitating procedures in confined and difficult to access anatomical regions. However, complications due to the reduced field-of-view provided by the surgical camera limit the surgeon's ability to self-localize. Computer assisted interventions (CAI) can help a surgeon by integrating additional information. For example, overlaying pre- and intra-operative imaging with the surgical console can provide a surgeon with valuable information which can improve decision making during complex procedures [13]. Integrating this data is a complex task and involves understanding relations between the patient anatomy, operating instruments and surgical camera. Segmentation of the instruments in the camera images is a crucial component of this process and can be used to prevent rendered overlays from occluding the instruments while providing crucial input to instrument tracking frameworks [12, 18].

Segmentation of surgical tools from tissue backgrounds is an extremely difficult task due to lighting challenges such as reflections, shadows and occlusions such as smoke and blood. Early methods attempted to simplify the problem by modifying the appearance of the instruments [19]. However, this complicates clinical application of the technique as sterilization can become an issue. Segmentation of the instruments using natural appearance is a more desirable approach as it can be applied directly to pre-existing clinical setups. However, this defines a more challenging problem. To solve it, previous work has relied on machine learning techniques to model the complex discriminative boundary. Approaches based on Random Forests [20], maximum likelihood Gaussian Mixture Models [18] and Naive Bayesian classifiers [21], all trained on color features, have been applied. More recently, the state-of-the-art has increasingly been defined by Fully Convolutional Networks (FCNs), such as the FCN-8s model [22] adapted for the task of binary segmentation of robotic tools [23] and U-Net [24] which was used for both binary and instrument part segmentation [25]. In order for segmentation approaches based on supervised training of neural networks to successfully generalize, a considerable amount of annotated data is required [15]. At the same time, creating a segmentation dataset with high resolution fine annotations is an extremely time-consuming and costly process [15] [1].

Recently, unsupervised and self-supervised methods were introduced to make use

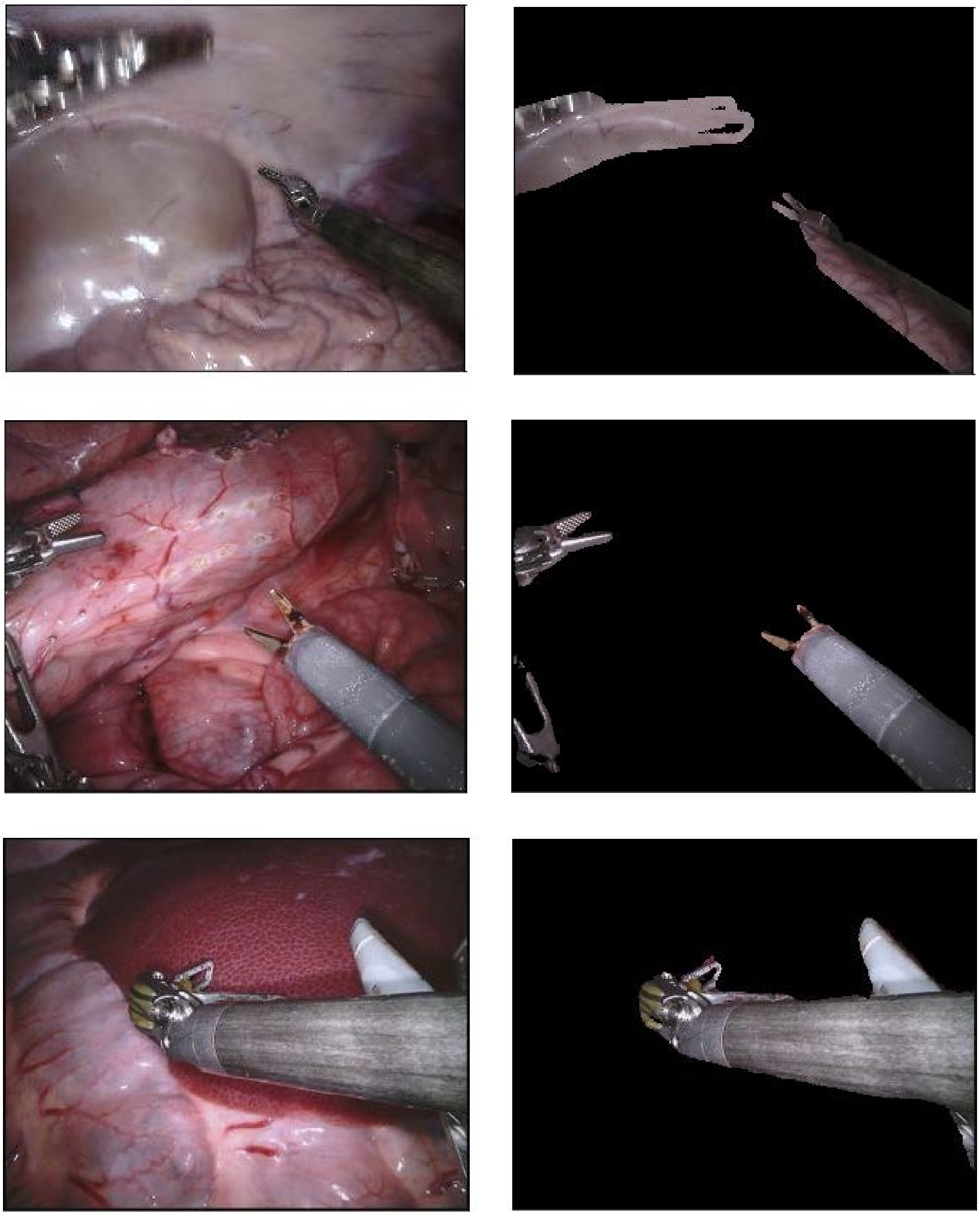


Figure 4.1: Top row shows an example of synthetic annotation acquired using forward kinematics model that was used for training of our method. As it can be seen, it barely captures the actual tool due to errors. Two bottom rows show example segmentations delivered by our method on a random images from Endovis 2017 dataset that were not used for training.

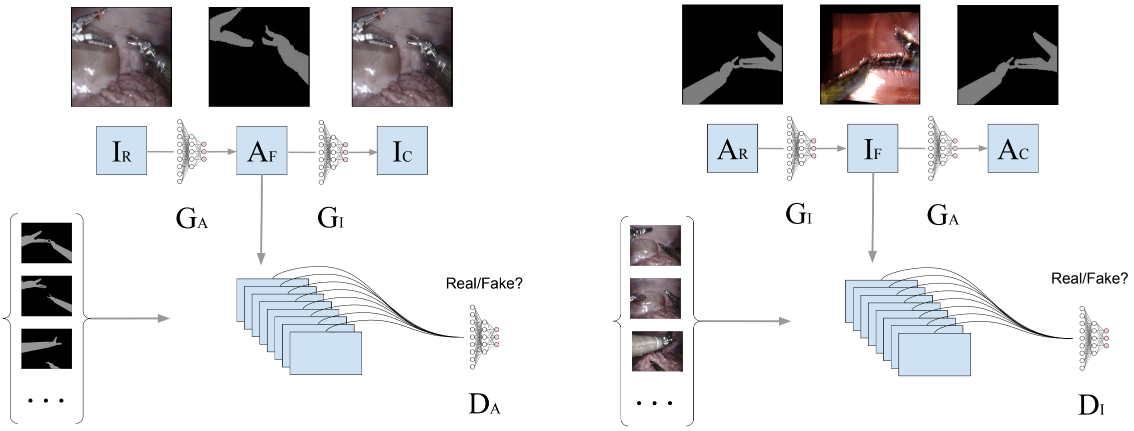


Figure 4.2: Main stages of our approach. During training, we learn four mappings which are represented by convolutional neural networks: generators G_A and G_I which map images to annotations and annotations to images respectively; discriminators D_A and D_I which learn to differentiate between real and generated annotations and images respectively. I_R and A_R stand for real images and annotations. I_F and A_F stand for images and annotations generated by networks. I_C and A_C stand for images and annotations generated from A_F and I_F respectively. They are used in cycle-consistency loss term. Loss also includes adversarial terms that forces discriminator to differentiate between real samples and the ones generated by the generators, and the generators to fool the discriminators. After the training, we use G_A to acquire segmentations of unseen images.

of a great amount of unlabeled images of surgical instruments which can be collected with little or no effort: Ross et al. [39] proposed a method, which allows to pre-train segmentation models on recoloring task, greatly reducing the number of labeled images necessary for supervised learning; Rocha et al. [40] proposes an optimization method to obtain corrected labels to train a binary segmentation model despite imprecise kinematic model which sometimes results in poor results when the approximate labels are located far from the actual position of a surgical tool. Apart from instrument segmentation domain, Mahmood et al. [41] trained a model for multi-organ nuclei segmentation with synthetic and real data, Pfeiffer et al. [42] generated synthetic dataset for liver segmentation and showed promising results.

One way to acquire segmentation masks automatically is to use kinematic model of a robot and CAD model of the tools and project them onto the image plane of camera. Unfortunately, since the kinematic model is imprecise, the generated projections exhibit errors and cannot be directly used for supervised training. Although a lot of effort was made in improving supervised training of instrument segmentation models [5] [25] [23], a method that is able to successfully use this kind of data for training can potentially be supplied with unlimited amount of automatically generated data.

In this paper, we introduce an approach for binary surgical tool segmentation which does not need any manual annotation and only uses labels generated with imprecise kinematic model. The problem is posed as image-to-image translation [43] where we need to convert an image of surgical scene from RGB representation to semantic labels representation. Due to errors in the generated annotations that are caused by the imprecise kinematics, we are not able to train a segmentation in a supervised setting using acquired image/annotation pairs. Although we lack direct supervision in the form of image and annotation pairs, we employ set-level supervision: we are given a set of surgical images and a set of generated annotations which were created automatically without any manual annotation (see Fig. 4.4). By using this approach, we learn the mapping from surgical images to binary tool segmentation which is competitive with supervised algorithms that use expensive manually annotated data. This will allow segmentation models to be trained on larger amount of data and generalize better.

4.2 Method

4.2.1 Data Generation

In our work we are not using images with manually created labels similar to Endovis 2017 [1] for training and, instead, rely only on labels generated with imprecise kinematics. We record image sequences of surgical procedures and corresponding annotation masks. The masks are generated by rendering CAD models of each instrument that is attached to a da Vinci Xi system using joint encoder values that are synchronized to the video feed. Rigid body transforms between the robot base frame and the instruments and the camera are computed using forward kinematics from the DH parameters of the robot. Camera calibration is also acquired from the system. Inaccuracies exist between the true instrument and camera poses due to unknown hand-eye calibration transforms and errors from slack and tension in the cable driven arms of the da Vinci [12].

Overall, three video sequences of different procedures were collected resulting in 6 thousand frames and annotations. As mentioned previously, annotations exhibit errors similar to example presented in Fig. 4.1. The process of the data creation is fully automatic and does not involve any manual annotation and, therefore, can be used to generate a dataset of potentially unlimited size (see Fig. 4.4).

Collected video sequences contain 3 different types of robotic surgical instruments: Large Needle Driver, Prograsp Forceps, Bipolar Forceps. An example image sample from the dataset is shown in Fig. 4.1. Although we were not able to collect video sequences featuring all instruments [1] because their CAD models are not available, we noticed that our approach successfully generalizes to previously unseen instruments.

4.2.2 Set-Level Supervision

Since the annotations that were automatically generated for our collected images contain errors, we can not use supervision on the level of image/annotation pairs during training. Instead, we use set-level supervision: we propose an unpaired GAN-based approach that learns mappings between surgical images (I) and annotation masks (A) and enforces them to be cycle consistent. We use cycleGAN [43] approach to learn a mapping $G_A : I \to A$ between surgical images and corresponding annotation masks that generalizes to previously unseen surgical images.

Overall, our approach employs four networks: G_I (segmentation labels to surgical image generator), G_A (surgical image to segmentation labels generator), D_A (discriminator network of G_A), and D_I (discriminator network of G_I). The final objective [43] consists of two adversarial loss terms $\mathcal{L}_{\text{GAN}}$ and cycle consistency loss term $\mathcal{L}_{\text{cyc}}$. After the training is done, we use G_A as our segmentation model and evaluate it on previously unseen images to assess its segmentation performance.

First, we introduce an adversarial loss [43] that forces our segmentation model G_A to translate surgical images to realistic segmentation masks by making them look similar to annotations that we collected from kinematics model:

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G_A, D_A) = {} & \mathbb{E}_{a \sim p_{\text{data}}(a)}[\log D_A(a)] \\ & + \mathbb{E}_{i \sim p_{\text{data}}(i)}[\log(1 - D_A(G_A(i))]\end{aligned} \tag{4.1}$$

Where D_A is a discriminator network that learns to distinguish between segmentations generated by our model G_A and real segmentations. Distributions $p_{\text{data}}(i)$ and $p_{\text{data}}(a)$ are approximated by surgical images and annotation samples collected in the prevous step using kinematics model. However, optimizing this objective alone does not guarantee that a meaningful mapping G_A will be learnt since there are infinitely many mappings that will satisfy this criteria [43].

To make our problem more well-defined we follow [43] and add a constraint on translation to be cycle-consistent: we introduce a reverse mapping $G_I : A \to I$ from surgical images to corresponding segmentation masks (Fig. 2) and make G_I and G_A to be inverses of each other. Note, that the reverse mapping is introduced only to train our segmentation model G_A more effectively [43].

Second adversarial term of loss is introduced to force the distribution of surgical images generated by the network G_I to be similar to that of the real surgical images:

$$\begin{aligned}\mathcal{L}_{\mathrm{GAN}}(G_I, D_I) = {} & \mathbb{E}_{i \sim p_{\mathrm{data}}(i)}[\log D_I(i)] \\ & + \mathbb{E}_{a \sim p_{\mathrm{data}}(a)}[\log(1 - D_I(G_I(a))] \end{aligned} \tag{4.2}$$

where the mapping G_I is constrained to generate surgical images from real segmentation masks so that they look like real surgical images. Discriminator D_I is trying to differentiate between real and generated surgical images. This term introduces competition between generator and discriminator networks, allowing them to improve during training as a result of the competition [43].

As previously mentioned, in order to make our mapping more well-defined and learn a better segmentation model we also introduce a cycle consistency loss:

$$\begin{aligned}\mathcal{L}_{\mathrm{cyc}}(G_A, G_I) = {} & \mathbb{E}_{i \sim p_{\mathrm{data}}(i)}[G_I(G_A(i)) - i_1] \\ & + \mathbb{E}_{a \sim p_{\mathrm{data}}(a)}[G_A(G_I(a)) - a_1] \end{aligned} \tag{4.3}$$

This term adds an additional regularization by making G_I and G_A inverse functions of each other. Specifically, it ensures that $G_I(G_A(i)) \approx i$ (forward cycle consistency) and $G_A(G_I(a)) \approx a$ (backwards cycle consistency).

The full objective for our set-level supervised learning can be written as:

$$\begin{aligned}\mathcal{L}(G_A, G_I, D_A, D_I) =& \mathcal{L}_{\text{GAN}}(G_A, D_A) \\ &+ \mathcal{L}_{\text{GAN}}(G_I, D_I) \\ &+ \lambda \mathcal{L}_{\text{cyc}}(G_A, G_I)\end{aligned} \tag{4.4}$$

Where λ is a hyperparameter. We aim to solve:

$$G_A^*, G_I^* = \arg \min_{G_A, G_I} \max_{D_A, D_I} \mathcal{L}(G_A, G_I, D_A, D_I) \tag{4.5}$$

4.2.3 Edge Consistency

Applying the aforementioned method as-is allows us to learn the mapping G_A that acts as a segmentation model and delivers realistically looking segmentation masks but upon inspection, they are completely unaligned with instruments present in the input image. Intersection over union accuracy measure is also very low, which means that the segmentation has little or no overlap with tools located in the image. This motivates us to introduce additional constraints so that generated annotations are more aligned with instruments located in the surgical images.

We replace U-Net [24] architecture that we used for generator networks G_I and G_A with a network based on deep residual connections which give us much better results [26] [43]. We hypothesize that an implicit regularization that residual con-

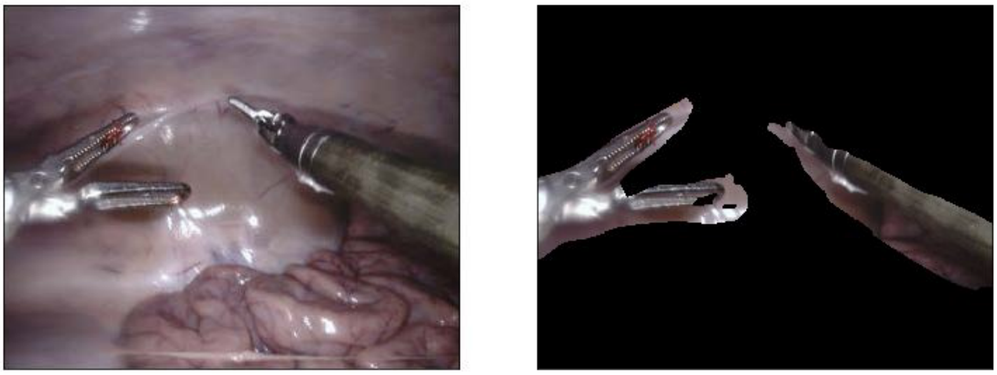


Figure 4.3: An example of segmentation results delivered by a deep residual model without edge consistency loss term. As it can be seen, while the shape of the segmentation looks realistic, it is not aligned with the edges of the tools. Since the segmentation result still has a great overlap with the tools, the intersection over union score is good but applications involving augmented reality and tracking of tools require the segmentation algorithm to be more precise along the borders.

nections provide [44] allows us to acquire segmentation results that are more aligned with the input image.

However, the problem is not completely solved: while the intersection over union score is good, the predictions delivered by the method are not precise along the borders (See Fig. 4.3). In order for the segmentation method to be useful for augmented reality applications and tool tracking, the segmentation results should be aligned with the borders of the actual tool [1]. Ideally, we would want to have consistent edges in the image and generated annotation (See Fig. 4.5). Inspired by a similar problem in the field of image matting [45] [46], we add the edge consistency term [45] to our loss:

$$\mathcal{L}_{\text{edge}}(G_A) = \mathbb{E}_{i \sim p_{\text{data}}(i)}[L_C(G_A(i), i)], \tag{4.6}$$

$$L_C(A, I) = \frac{\sum A_{mag}\left[1 - (I_x A_x + I_y A_y)^2\right]}{\sum A_{mag}} \tag{4.7}$$

where (I_x, I_y) and (A_x, A_y) are the normalized image and annotation gradients, and A_{mag} is the annotation gradient magnitude. Intuitively, the loss constraints the generator G_A to deliver segmentation masks that are more aligned with the image edges (See Fig. 4.5). Using both the deep residual network and edge consistency gives the best results (See Fig. 4.1).

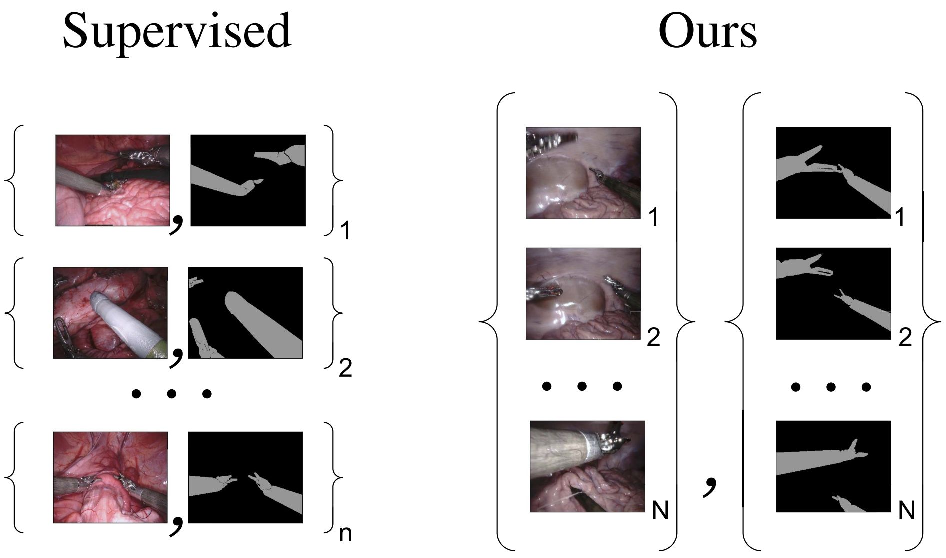


Figure 4.4: Supervised learning approach typically uses paired training data (left) consisting of training examples where every image has a corresponding manually annotated image. These datasets are expensive to collect and are usually of a small size. We instead consider unpaired training data (right), consisting of images and synthetically generated annotations using forward kinematics. This data can be automatically generated during medical procedures and can potentially be of unlimited size $N \gg n$.

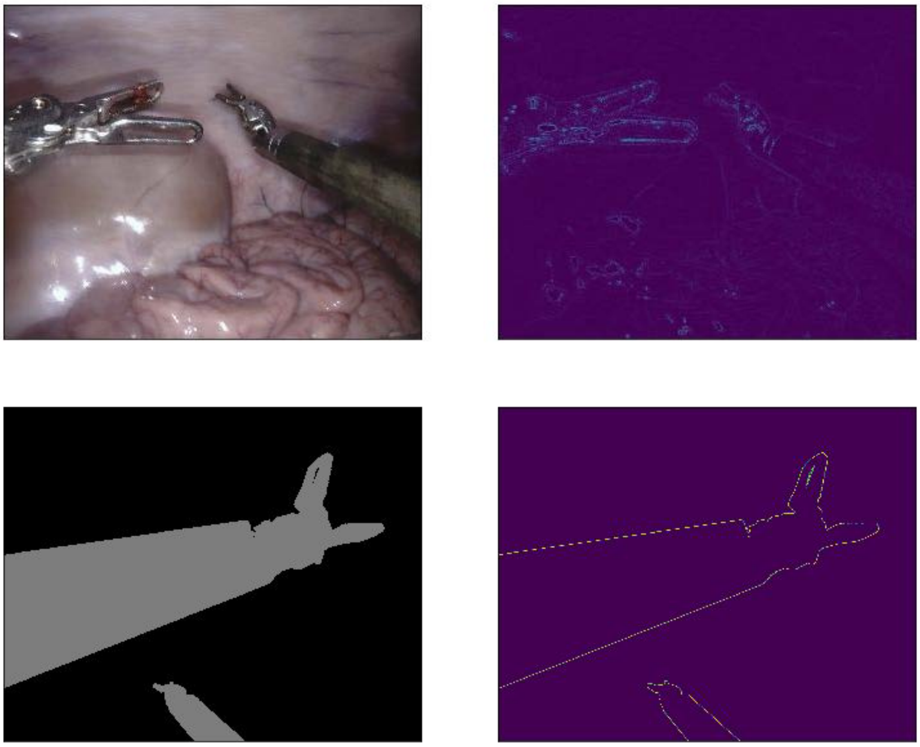


Figure 4.5: Examples of surgical image and annotation and their gradient magnitude images. The figures serves as a motivation for edge consistency loss: edges of the image should be aligned with the edges of the segmentation generated by the network. (Image is best viewed in the electronic version of the document)

4.3 Experiments and Results

4.3.1 Implementation Details

4.3.1.1 Network Architectures

The generator architecture G_I and our segmentation model G_A are both based on deep residual network architecture [26], since it was shown to work better for our task. First two layers of the networks subsample the input image by a factor of 4, followed by nine residual blocks and two upsampling layers with learnt filters that bring the output to the same size as the input image. The discriminator networks D_I and D_A have much simpler architectures based on PatchGANs [47] with three layers and fewer parameters, as suggested in [43]. All networks are implemented in a fully-convolutional fashion, which allows them to be applied to images of varying sizes [22].

In order to reduce memory consumption and be able to store four networks in GPU memory at the same time during training we had to resort to training with batch size one. Since training with small batch size with batch normalization is known to be unstable [38], we are employing instance normalization layers instead [48].

4.3.1.2 Training details

All the networks in our work were trained from scratch, starting with randomly initilized weights. We perform the optimization with Adam optimizer [29] with batch size one and learning rate of 0.0002. Overall, we train for 20 epochs with fixed learning rate and then linearly decay it to zero for another 20 epochs.

4.3.2 Accuracy measures

There are three commonly used accuracy measures for assessment of image segmentation models [49]:

1. Overall Pixel accuracy

2. Per-Class accuracy

3. Jaccard Index (Intersection-over-union)

In order to describe accuracy measures we define confusion matrix $\mathbf{C}$, which contains predictions for the whole image segmentation dataset $\mathbf{D}$:

$$\mathbf{C}_{ij} = \sum_{I \in \mathbf{D}} \left| z \in \mathbf{I} \ such \ that \ S^{I}_{gt}(z) = i \ and \ S^{I}_{ps}(z) = j \right|$$

Where $S^{I}_{gt}(z)$ is the ground-truth label of pixel z in image $\mathbf{I}$, $S^{I}_{ps}(z)$ is a label predicted by a particular algorithm and $|A|$ is the cardinality of the set A. We define $\mathbf{G}_i = \sum_{j=1}^{L} C_{ij}$, where L is the number of classes and $\mathbf{P}_j = \sum_i \mathbf{C}_{ij}$

Overall Pixel accuracy measures number of correctly classified pixels:

$$OP = \frac{\sum_{i=1}^{L} \mathbf{C}_{ii}}{\sum_{i=1}^{L} \mathbf{G}_i}$$

One significant limitation of this measure is its bias in the presence of very imbalanced classes [49]. If a dataset has one class that is more present than others and a segmentation model classifies it correctly, while making mistakes on other smaller classes, the value of the measure will not sufficiently represent that.

Per-Class accuracy does the same measurement as Overall Pixel accuracy but solves its problem with unbalanced classes by scaling results of from each class [49]:

$$PC = \frac{1}{L}\sum_{i=1}^{L}\frac{\mathbf{C}_{ii}}{\mathbf{G}_i}$$

While it solves the problem, another weakness can be observed: if a large background class is present, one achieves a better score by labeling object classes correctly while making more errors in labeling the background class [49].

Jaccard Index (Intersection-over-union) measures intersection over union for each class and reports the average between all classes:

$$JI = \frac{1}{L}\sum_{i=1}^{L}\frac{\mathbf{C}_{ii}}{\mathbf{G}\ _i + \mathbf{P}_i - \mathbf{C}_{ii}}$$

It solves the problem of previous two measures and currently is a main measure of semantic segmentation accuracy for PASCAL VOC challenge [50] and Endovis 2017 Robotic Instrument Segmentation Challenge [1] which we use to assess our method. In order to be able to compare our method with other methods on Endovis 2017 challenge we report Intersection over Union accuracy measure of our method on test dataset.

	NCT	UB	BIT	MIT	SIAT	UCL	TUM	Delhi	UA	UW	Ours	+Edge
Dataset 1	0.784	0.807	0.275	0.854	0.625	0.631	0.760	0.408	0.413	0.337	0.692	0.727
Dataset 2	0.788	0.806	0.282	0.794	0.669	0.645	0.799	0.524	0.463	0.289	0.735	0.769
Dataset 3	0.926	0.914	0.455	0.949	0.897	0.895	0.916	0.743	0.703	0.483	0.721	0.755
Dataset 4	0.934	0.925	0.310	0.949	0.907	0.883	0.915	0.782	0.751	0.678	0.752	0.782
Dataset 5	0.701	0.740	0.220	0.862	0.604	0.719	0.810	0.528	0.375	0.219	0.778	0.794
Dataset 6	0.876	0.890	0.338	0.922	0.843	0.852	0.873	0.292	0.667	0.619	0.793	0.815
Dataset 7	0.846	0.930	0.404	0.856	0.832	0.710	0.844	0.593	0.362	0.325	0.686	0.694
Dataset 8	0.881	0.904	0.366	0.937	0.513	0.517	0.895	0.562	0.797	0.506	0.787	0.815
Dataset 9	0.789	0.855	0.236	0.865	0.839	0.808	0.877	0.626	0.539	0.377	0.673	0.726
Dataset 10	0.899	0.917	0.403	0.905	0.899	0.869	0.909	0.715	0.689	0.603	0.706	0.727
Mean IOU	0.843	0.875	0.326	0.888	0.803	0.785	0.873	0.612	0.591	0.461	0.732	0.760

Table 4.1: Quantitative results of our method and comparison with supervised methods in binary segmentation of robotic tools [1].

4.3.3 Dataset

We test our method on the EndoVis 2017 Robotic Instruments dataset [1]. There are 8 high resolution (1280 × 1024) sequences with 225 frames each in the training dataset [1]. As mentioned previously, we did not use the training dataset of Endovis 2017 and, instead, trained our method on our data acquired with imprecise kinematics model. Each pixel is labeled as either tool or background. There are 10 75-frame sequences in the test dataset that features 7 different robotic surgical instruments [1]. Samples from the dataset and segmentations delivered by our algorithm are depicted in Fig. 4.1.

The dataset is very challenging and even some of the supervised image segmentation methods were struggling to achieve good performance as it can be seen from the Table. 4.1. At the same time it provides a good indication of whether our method can generalize well to unseen surgical scenes and instruments. After training our segmentation model on collected images and annotations generated with imprecise kinematics model, we evaluated it on the test set of the Endovis dataset.

4.3.4 Quantitative Study and Results

We used data from ten different test sequences of Endovis 2017 dataset to evaluate our trained instrument segmentation network. To assess generalizability of the developed algorithm, we also payed attention on how our method performed segmentation with instruments that were not represented in our collected dataset: our method successfully segmented previously unseen instrument (See bottom row of Fig. 4.1). Table 4.1 summarizes the quantitative results of the testing with intersection-over-union metric.

A comparative analysis with supervised segmentation methods that participated in the challenge was performed and our method, as can be seen from the table, outperforms four out of ten methods. We are the first ones to test a self-supervised method on challenging Endovis 2017 competition while outperforming some of the supervised methods that reported their results.

4.4 Conclusion and Future Work

Automated training of accurate instrument segmentation models for surgical procedures has the potential to completely eliminate costs associated with manual creation of datasets and can greatly affect the field by improving the segmentation performance and robustness of segmentation models by employing abundant unlabeled data. In this work we address the problem of training a segmentation model without direct supervision where images and inaccurate labels are generated automatically, therefore, eliminating the need for dataset creation.

We propose an approach that allows the instrument segmentation network to be trained on images with synthetically generated annotations with errors. The problem is posed as an unpaired image-to-image translation task. This way we are able to enforce set-level supervision between sets of surgical images and annotations. This approach performs on par with some standard supervised approaches tested on challenging Endovis 2017 dataset.

In the future work, we plan on adapting this approach to multi-class instrument segmentation and instance segmentation since this data can be easily automatically generated in a similar way to how we generated binary masks. Potentially, other types of mappings can also be learnt without direct supervision that can be very useful for surgical scene analysis and pose estimation like pixel-wise depth estimation and surgical tools landmarks detection. On the other hand, since our approach still performs worse than some of the supervised methods in terms of accuracy, a better segmentation network architecture can be used to close this gap in accuracy and completely eliminate the need for manual segmentation dataset creation.

Chapter 5

Unsupervised Semantic Image Segmentation with Stylegan and CLIP

In this chapter, we deviate from the common assumption that an abundance of annotated or synthetic data is available for training. First, we consider representation learning in a fully unsupervised fashion, and we demonstrate that generative adversarial learning works best for this purpose. Here, we show that this leads to the discovery of semantically meaningful regions from unlabeled data. Our approach yields state-of-the-art performance on multiple datasets. Next, we study the ability of recently introduced multimodal approaches like CLIP to assign text labels to our discovered segmentation regions. At the end, we present a model that is able to not only discover segmentation regions automatically but also assigns text labels them.

5.1 Introduction

The development of deep convolutional neural networks [26, 51, 52] has fueled remarkable progress in semantic segmentation and pushed state-of-the-art on a variety of datasets [53–55]. While the introduced methods show impressive results, their applicability is limited by a very slow process of image annotation [15]. This issue is more pronounced in the medical imaging community where a domain knowledge is required to perform annotations. Consistent labeling can be a difficult task for a team of annotators working on the same dataset as most semantic classes lack clear visual boundaries. This can lead to some heterogeneity in the labels which can affect the performance of a model. At the same time most deep learning approaches require a considerable amount of training data to reach their best performance [26]. In this paper, we introduce a method that does not require any labeling, allows to discover consistent semantic regions across images, outperforms semi-supervised methods that require a small number of annotations and is competitive with fully supervised image segmentation methods.

In our work we compare our method against semi-supervised learning approaches (SSL) which use a small number of annotations. This serves as a very strong baseline for our approach and we show that we outperform recent state-of-the-art methods [56, 57] even though we do not use any annotations. Our method allows to suggest consistent semantic classes and we show that on some datasets they coincide with annotations provided by human which allows us to measure its accuracy. As it can be seen our method allows to discover high quality semantic regions that coincide with human defined regions (see Fig. 5.1) and also suggest consistent semantic regions that might be hard for human to discover and label (see Fig. 5.2). Additionally, we introduce a method that allows us to use natural language to discover certain rare semantic regions like beard, glasses and hats (see Fig. 5.6). Finally, we show how we can automatically classify each of the discovered semantic regions and assign a prompt defined in a natural language to each of them (see Fig. 5.5). To demonstrate applicability of our method across other domains, we carry out experiments on an eye segmentation dataset [4] and show that our discovered semantic regions replicate human annotations (see Fig. 5.8) and that the accuracy of our provided segmentation masks is competitive with fully supervised methods.

5.2 Method

Our approach is based on StyleGAN2 [58] a very powerful generative model capable of generating realistic samples of objects of interest once it has been trained on a related dataset (Section 5.2.1). In our work we either use a pretrained models in case of human faces, cats, dogs and cartoons or train a model from scratch on a desired dataset in case of eye segmentation problem [4].

We show that by clustering in the feature space of trained StyleGAN2 model, we can discover meaningful semantic regions that are consistent across different samples of the model (Section 5.2.2).

To discover rare semantic classes that were not found during clustering since they appear relatively rarely in samples of StyleGAN2, we apply recently introduced approach for image manipulation [9, 10] using text prompts with CLIP [11]. By manipulating latent vectors of sampled images, we make a desired semantic region to appear in almost every sample, which allows it to appear as one of the clusters (Section 5.2.3).

To conclude the process and attribute each semantic region with a prompt defined in a natural language, we modify the original CLIP [11] image encoder to accept image regions instead of a whole image. Then we encode each of the discovered semantic regions and classify them (Section 5.2.4).

Once a StyleGAN2 model is modified to deliver samples along with segmentation masks, we apply a simple knowledge distillation procedure by training a segmentation network on synthetic images and segmentation masks and show that it generalizes to real images (Section 5.2.5).

5.2.1 StyleGAN2

StyleGAN2 is used in our method as a generative backbone as it is able to generate images with high quality, gives us access to feature maps of generated images that we use for clustering and allows generated images to be manipulated which we use to discover additional semantic classes. Sampling is performed by drawing from a normal distribution $z \in Z$. A separate mapping network maps z to an intermediate latent vector $w \in W$. w is then transformed into k vectors that are used by different layers of generator network of StyleGAN2 (see Fig. 5.3). StyleGAN2 gradually generates feature maps of higher resolution as they get upsampled and processed by successive layers of the generative model.

5.2.2 Clustering

During clustering, we generate N samples and save the generated images as well as corresponding feature maps from a certain layers of StyleGAN. We mostly use feature maps from the seventh or ninth layer of the generative model. Using earlier layers of StyleGan results in more coarse but more semantically meaningful clusters. Later layers result in more fine clusters with less semantic meaning. We also performed experiments with concatenated features from different layers but did not notice any improvement. Feature maps have dimensionality of either $N \times 64 \times 64 \times 512$ or $N \times 128 \times 128 \times 256$ where the first number represents a number of samples the second and third numbers represent the spatial resolution and the last number represents the dimensionality of the feature space. We flatten the acquired feature maps and end up with $N * 64 * 64$ feature maps of size 512 or with $N * 128 * 128$ feature maps of size 256. After that we perform clustering by minimizing the following loss function:

$$J = \sum_{n=1}^{\bar{N}} \sum_{k=1}^{K} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$$

Where $\boldsymbol{x}_n$ represents a single data point of size 512 or 256, $r_{nk} \in 0, 1$ is a binary indicator variable, where $k = 1, \ldots, K$ describes which of the K clusters the data point $\boldsymbol{x}_n$ is assigned to. $\bar{N}$ is equal to $N * 128 * 128$ or $N * 64 * 64$ depending on the layer of choice. We search for set of values $\{r_{nk}\}$ and $\{\boldsymbol{\mu}_k\}$ that minimize J. To solve this problem we use K-means clustering algorithm [59]. Discovered classes can be seen in Fig. 5.2. After the clusters are discovered we notice that new samples have the same semantic regions if we assign their features to the closest cluster. This gives us a way to generate image and corresponding segmentation masks at the same time with minimal additional computational overhead.

5.2.3 Class Discovery with Image Manipulation

Some classes that are rarely present in samples of StyleGAN like hat, beard or glasses are not always discovered by our method as is. In order to make desired semantic classes more present in samples that we use for clustering, we update a latent code of every image so that it contains a desired attribute. Let $w \in W$ denote a latent code of a generated sample, and F(w) the corresponding generated image (where F is a generator network of StyleGAN). We employ a recently introduced method [9,10]: we collect a number of latent vectors and corresponding generated images and classify generated images with CLIP [11] given a natural language prompt like “a person with a beard”. Given a set of pairs $\{(w_i, b_i)\}$, where w is a latent vector and b is a binary variable indicating if the generated images contains desired attribute, we learn a manipulation direction G, such that $F(w + \alpha G)$ yields an image where that attribute is introduced or amplified. The manipulation strength is controlled by α. The method depiction can be seen in Fig. 5.4. An example of latent code manipulation along the beard direction can be seen in Fig. 5.9 and discovered corresponding semantic regions are depicted in Fig. 5.6.

5.2.4 Cluster Classification

Once all the desired clusters are discovered, we need to assign a semantic class to each of them. In order to do so, we use pretrained text encoder and image encoder of the CLIP model [11]. We use text encoder as is and encode desired classes given corresponding text prompts like “hair“, “forehead“ etc (see Fig. 5.5). As for the image encoder we use a model based on residual neural network [11,26]. Our choice is motivated by the fact that this model can be modified to deliver embedding with bigger spatial resolution by removing downsampling layers and adding dilation factors to certain convolutional layers [53, 55]. We observed that bigger spatial resolution gives better results for smaller clusters. Once we get embeddings for each pixel of every image, we average them per each cluster over many images. Averaging over many images also gives better results according to our experiments. Finally, we compute the dot products between embedding of each cluster and each text prompt: we assign the cluster to a class that resulted in a biggest dot product value. This way we use original clusters as region proposals that we classify.

5.2.5 Knowledge Distillation

Once we are able to generate synthetic images and corresponding annotations, we need a way to get the same segmentation masks for real images. Inspired by recent work in knowledge distillation for image manipulation with StyleGAN [60] we generate a synthetic dataset with images and corresponding segmentation masks and train a simple segmentation model on that dataset. We apply a segmentation method based on resnet-18 and dilated convolutions [55] and show that trained model also generalizes to real images as it can be seen in Fig. 5.7

5.3 Datasets and Results

For faces, we evaluate our model on CelebA-Mask8 dataset [2], which contains 8 part categories and compare our method to state-of-the-art semi-supervised methods [56, 57]. As can be seen in the Table 5.1 our method achieves state-of-the-art results. In order to evaluate our method on face images that contain a bigger set of labels we test our method on CelebAMask-HQ dataset [3] segmentation face parsing dataset that contains 19 semantic classes. To simplify the dataset we merge left and right eye classes into one and do similar simplification for ears. We also eliminate earring and necklace attributes as StyleGAN struggles to generate these attributes realistically which results in poor performance. The results compared to fully supervised method can be found in Table 5.2. As it can be seen our method is performing worse than the fully supervised method but delivers overall good results. We also collect a small dataset where we label beard in order to test the accuracy of our method on this discovered semantic class. The results compared to fully supervised method can be found in Table 5.4. To test if our method is able to generalize to completely different domains, we use our method to perform eye segmentation on OpenEDS [4] dataset. We train StyleGAN2 on images of the dataset from scratch and apply our algorithm to the trained model. As it can be seen, overall iris, pupil and sclera were discovered which coincides with semantic classes of the original dataset (see Fig. 5.8). Moreover, our algorithm achieves a segmentation performance which is very close to the fully supervised method in this case, since the semantic classes are simpler (see Table 5.3).

Method	DatasetGAN [57]	semanticGAN [56]	Ours
Number of Manually Annotated Images Used	16	30	**0**
Mean IOU	70.01	69.02	**73.1**

Table 5.1: Results of our algorithm on CelebA-Mask8 Dataset [2] in comparison to semi-supervised methods. As it can be seen our method outperforms them and does not need any manual annotations.

Method	DRN [55]	Ours
Number of Manually Annotated Images Used	All	**0**
Mean IOU	**70.5**	62.5

Table 5.2: Results of our algorithm on CelebAMask-HQ [3] in comparison to fully supervised method. The dataset has additional face semantic classes compared to CelebA-Mask8 Dataset [2].

5.4 Conclusion

We propose a powerful method for unsupervised learning that allows to discover meaningful and consistent semantic classes which mostly coincide with classes defined and labeled by human. Our method is successful in cases where it might be hard for human to define and consistently label semantic classes. We propose a way to derive semantic class annotations by defining a text prompt which allowed us to discover classes like beard that currently has no public annotations. We show that training a segmentation model on our generated synthetic images along with segmentation masks generalizes to real images and shows competitive results with fully supervised methods. Our method achieves state-of-the-art results as it performs better than recently introduced semi-supervised models.

Method	SegNet [4]	Ours
Number of Manually Annotated Images Used	All	**0**
Mean IOU	**84.1**	82.39

Table 5.3: Results of our algorithm on OpenEDS 2020 eye segmentation dataset [4] in comparison to fully supervised method. The dataset has 4 semantic classes. As it can be seen, our method has a very similar performance compared to fully supervised method as semantic classes are simpler in this dataset.

Method	DRN [4]	Ours
Number of Manually Annotated Images Used	All	**0**
Mean IOU	**88.8**	84.29

Table 5.4: Results of our algorithm on our custom beard segmentation dataset compared to fully supervised approach.

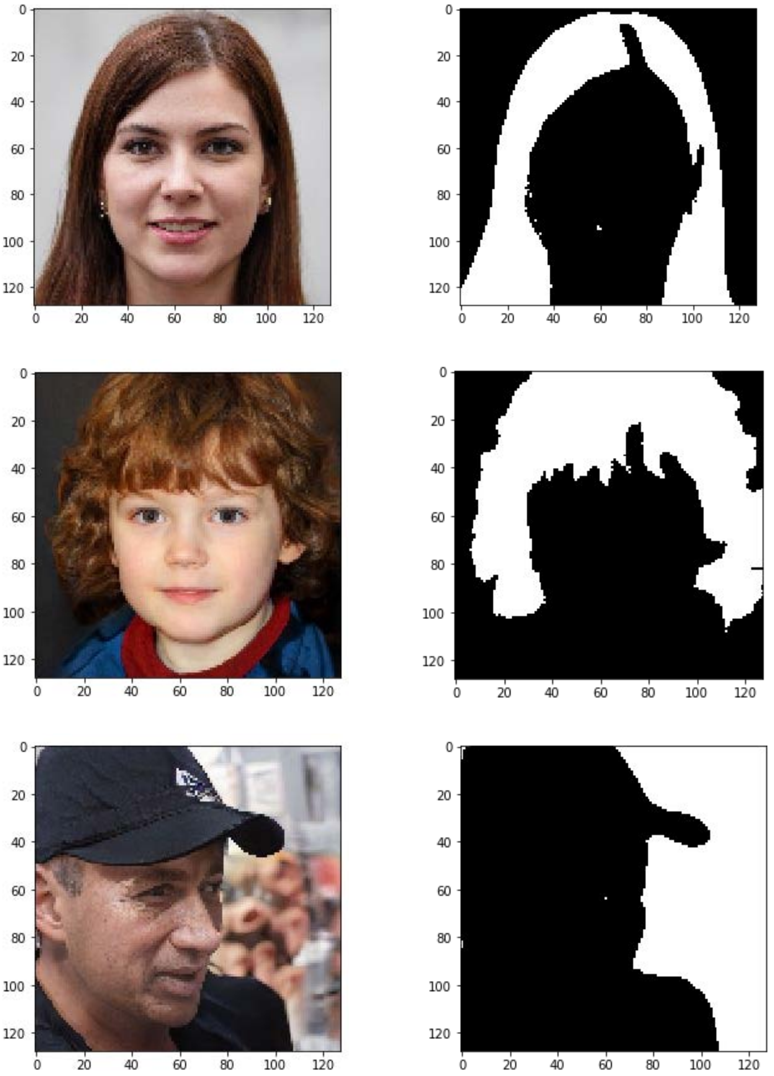


Figure 5.1: Example of synthetic image and annotation pairs created with our method for hair segmentation (first and second row) and background segmentation (third row).

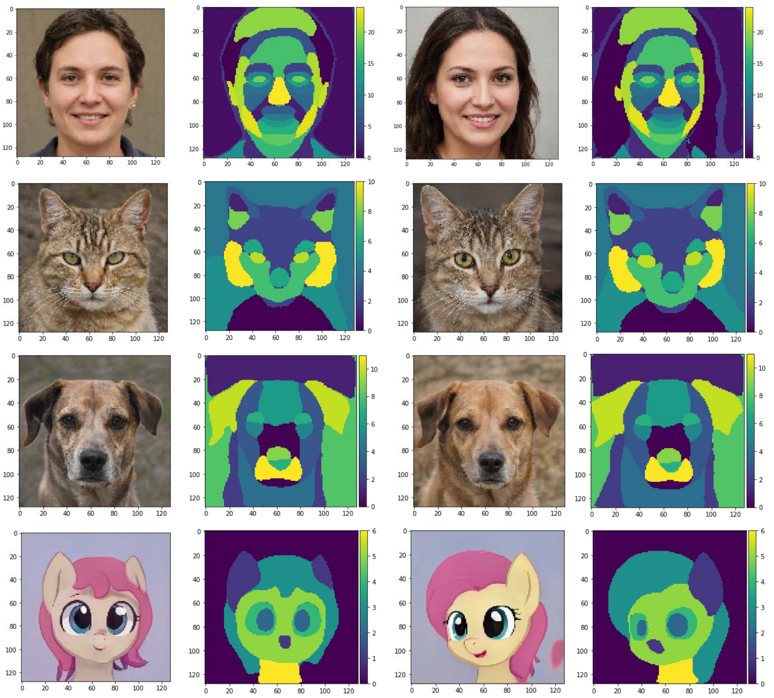


Figure 5.2: Synthetic dataset examples generated using our method with a stylegan model pretrained on Flickr-Faces-HQ (FFHQ) [6] dataset (first row), Animal faces (AFHQ) [7] dataset (second and third rows for cats and dogs) and a cartoon dataset [8]. Semantic regions proposed by the network are consistent across samples even though there is no clear visual border between most semantic classes: it is usually hard for human annotators to consistently label examples like this while our method works well.

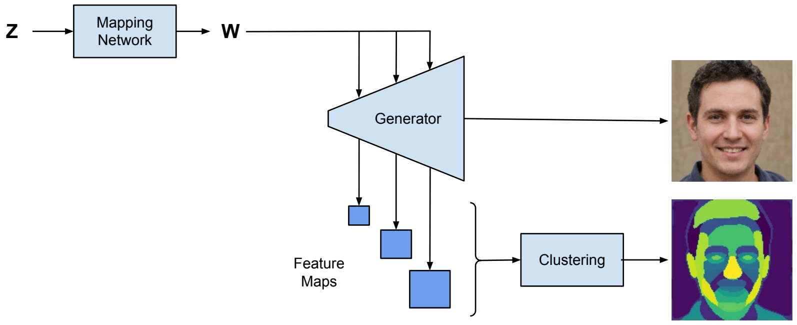


Figure 5.3: Generation of a synthetic dataset with semantic annotations for our approach. Stylegan consists of mapping network and generator networks which allow to create sythetic images. First we generate N images and save their intermediate feature maps produced by the generator network. Clustering allows us to find semantic regions of the generated images. After the clusters are found, a much bigger set of images is generated and using their feature maps we attribute each pixel to one of the previously discovered semantic clusters. A segmentation network is later on trained on the synthetic dataset.

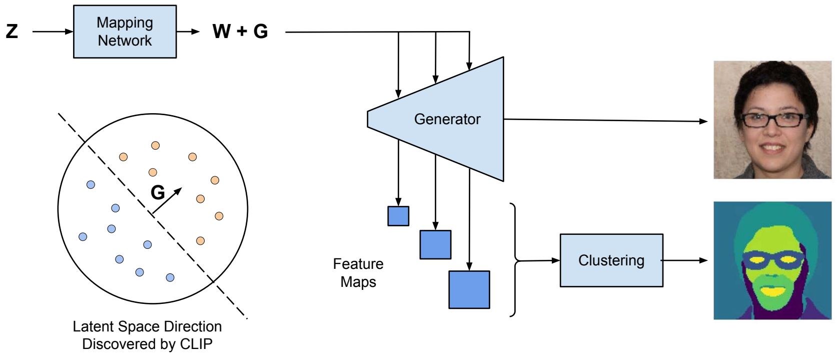
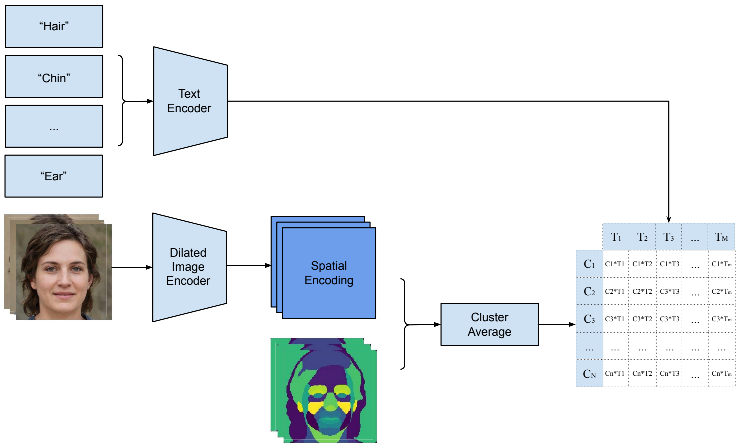


Figure 5.4: Generation of synthetic annotations for rare classes that were not discovered during clustering. First, a latent direction in the Stylegan is learnt that adds a desired semantic class to almost every generated sample using the method of [9, 10]. In this case, the vector G represents the a text promt “a person with glasses”. After that, almost every sample has the desired attribute and it naturally appears as one of the clusters. As it can be seen, semantic region representing glasses is indeed represented by one of the clusters.



	T_1	T_2	T_3	...	T_M
C_1	C1*T1	C1*T2	C1*T3	...	C1*T_m
C_2	C2*T1	C2*T2	C3*T3	...	C2*T_m
C_3	C3*T1	C3*T2	C3*T3	...	C3*T_m
...	...	...	...	...	...
C_N	Cn*T1	Cn*T2	Cn*T3	...	Cn*T_m

Figure 5.5: Classification of previously discovered clusters. Given a set of text prompts of desired classes and a set of generated images with corresponding clusters, we embed both text and image regions using pretrained text encoder and image encoders of CLIP [11]. After that we compute pairwise dot products between text and cluster embeddings. Each cluster is assigned to a text prompt that results in a biggest dot product value. For example, in a given set of images all clusters containing hair will be classified as "hair".

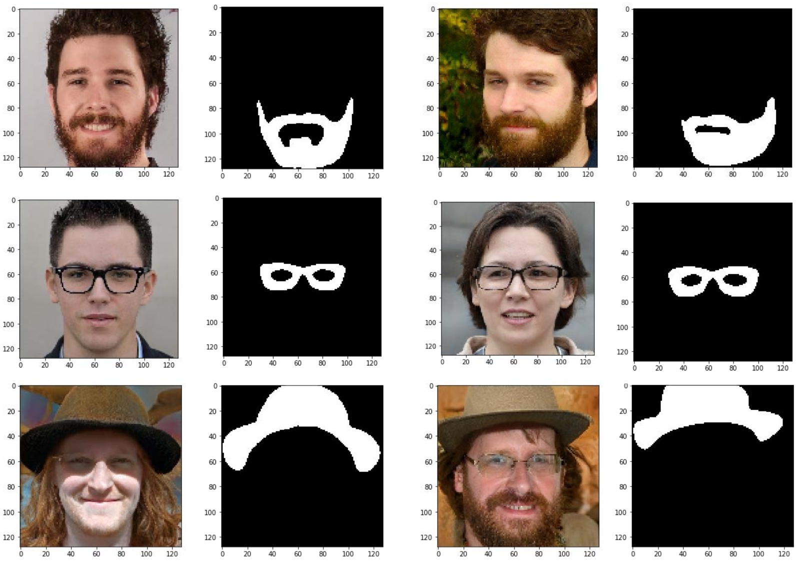


Figure 5.6: Example of synthetic image and annotation pairs created for rare classes with our method with CLIP [11] using text prompts “a person with a beard” (first row), “a person with glasses” (second row), “a person wearing a hat” (third row). Interestingly, the discovered class representing glasses does not include eye regions which is consistent with human annotations of glasses for some datasets like CelebAMask-HQ [3].

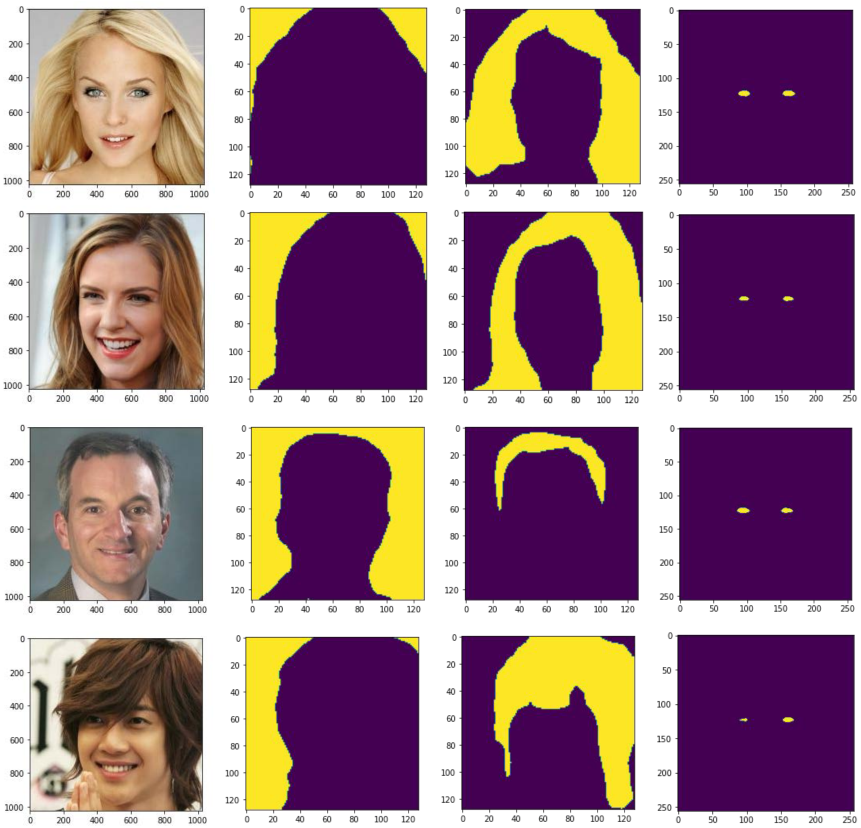


Figure 5.7: Segmentation results on test set of CelebAMask-HQ [3] delivered by our segmentation model on semantic classes with a increasing complexity: background, hair, eyes.

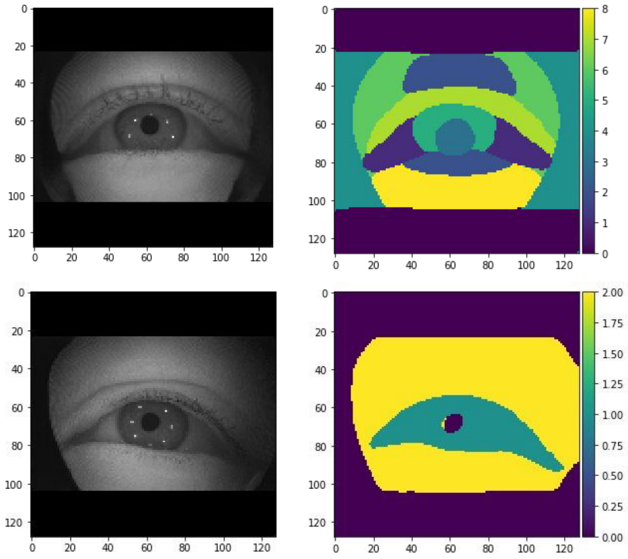


Figure 5.8: Example of semantic classes discovered by clustering features from different layers for OpenEDS eye segmentation dataset. As it can be seen, overall iris, pupil and sclera were discovered which coincides with semantic classes of the original dataset.

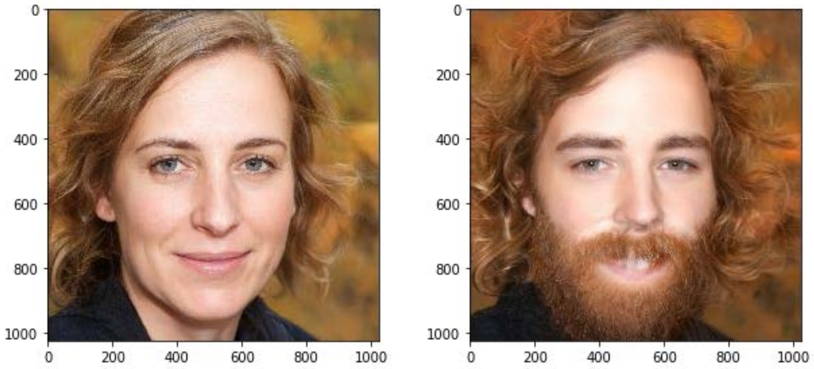


Figure 5.9: Example of latent code manipulation of generated image along the direction of beard attribute.

Chapter 6

Conclusions

This thesis focused on fast and data-efficient image segmentation, a fundamental task which has many applications including pose estimation of surgical tools in robotic surgery and eye tracking in head mounted displays.

In Chapter 2, we introduced deep dilated residual networks for the task of surgical surgical instrument segmentation, and demonstrated that it is possible to achieve highly accurate state-of-the-art results in both binary and part-based instrument segmentation tasks.

In Chapter 3, we departed from the common path of using manually designed networks for the task of instrument segmentation, and instead focused on learning an optimal architecture for the task of instrument segmentation. This led to state-of-the-art performance for both binary and part-based surgical instrument segmentation in terms of speed and accuracy.

In Chapter 4, we investigated the possibility of using only corrupted synthetic annotations for surgical instrument segmentation. We developed a new approach that is capable of learning accurate instrument segmentation model from corrupted synthetic annotations. We found that our method is comparable with fully supervised methods that use accurate human labels.

In Chapter 5, we asked whether it is possible to perform image segmentation without any annotations. First, we showed that generative adversarial learning can be used to learn meaningful representations that lead to the discovery semantically meaningful classes. Next, we showed that we are able to achieve state-of-the-art performance on multiple datasets. And finally, we found that by using recently introduced multimodal approaches like CLIP we are able to assign text labels to our discovered segmentation regions.

Bibliography

[1] M. Allan, A. Shvets, T. Kurmann, Z. Zhang, R. Duggal, Y.-H. Su, N. Rieke, I. Laina, N. Kalavakonda, S. Bodenstedt *et al.*, "2017 robotic instrument segmentation challenge," *arXiv preprint arXiv:1902.06426*, 2019.

[2] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.

[3] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "Maskgan: Towards diverse and interactive facial image manipulation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[4] C. Palmero, A. Sharma, K. Behrendt, K. Krishnakumar, O. V. Komogortsev, and S. S. Talathi, "Openeds2020: open eyes dataset," *arXiv preprint arXiv:2005.03876*, 2020.

[5] D. Pakhomov, V. Premachandran, M. Allan, M. Azizian, and N. Navab, "Deep residual learning for instrument segmentation in robotic surgery," in *Interna-*

tional Workshop on Machine Learning in Medical Imaging. Springer, 2019, pp. 566–573.

[6] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *arXiv preprint arXiv:1812.04948*, 2018.

[7] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8188–8197.

[8] "My little pony dataset," https://thisponydoesnotexist.net/, accessed: 2019-09-30.

[9] Y. Shen, C. Yang, X. Tang, and B. Zhou, "Interfacegan: Interpreting the disentangled face representation learned by gans," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[10] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, "Styleclip: Text-driven manipulation of stylegan imagery," *arXiv preprint arXiv:2103.17249*, 2021.

[11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," *arXiv preprint arXiv:2103.00020*, 2021.

[12] M. Allan, P.-L. Chang, S. Ourselin, D. J. Hawkes, A. Sridhar, J. Kelly, and

D. Stoyanov, "Image based surgical instrument pose estimation with multi-class labelling and optical flow," in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2015, pp. 331–338.

[13] R. H. Taylor, A. Menciassi, G. Fichtinger, and P. Dario, "Medical robotics and computer-integrated surgery," in *Springer handbook of robotics.* Springer, 2008, pp. 1199–1222.

[14] D. Pakhomov and N. Navab, "Searching for efficient architecture for instrument segmentation in robotic surgery," in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2020, pp. 648–656.

[15] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[16] S. B. Bhayani and G. L. Andriole, "Three-dimensional (3d) vision: does it improve laparoscopic skills? an assessment of a 3d head-mounted visualization system," *Reviews in urology*, vol. 7, no. 4, p. 211, 2005.

[17] A. M. Okamura, "Haptic feedback in robot-assisted minimally invasive surgery," *Current opinion in urology*, vol. 19, no. 1, p. 102, 2009.

[18] Z. Pezzementi, S. Voros, and G. D. Hager, "Articulated object tracking by render-

ing consistent appearance parts," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on.* IEEE, 2009, pp. 3940–3947.

[19] O. Tonet, T. Ramesh, G. Megali, and P. Dario, "Tracking endoscopic instruments without localizer: image analysis-based approach." *Studies in health technology and informatics*, vol. 119, pp. 544–549, 2005.

[20] D. Bouget, R. Benenson, M. Omran, L. Riffaud, B. Schiele, and P. Jannin, "Detecting surgical tools by modelling local appearance and global shape," *IEEE transactions on medical imaging*, vol. 34, no. 12, pp. 2603–2617, 2015.

[21] S. Speidel, M. Delles, C. Gutt, and R. Dillmann, "Tracking of instruments in minimally invasive surgery for surgical skill analysis," in *International Workshop on Medical Imaging and Virtual Reality.* Springer, 2006, pp. 148–155.

[22] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[23] L. C. Garcıa-Peraza-Herrera, W. Li, C. Gruijthuijsen, A. Devreker, G. Attilakos, J. Deprest, E. Vander Poorten, D. Stoyanov, T. Vercauteren, and S. Ourselin, "Real-time segmentation of non-rigid surgical tools based on deep learning and tracking," in *CARE Workshop (MICCAI)*, 2016.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for

biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[25] I. Laina, N. Rieke, C. Rupprecht, J. P. Vizcaíno, A. Eslami, F. Tombari, and N. Navab, "Concurrent segmentation and localization for tracking of surgical instruments," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2017, pp. 664–672.

[26] 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.

[27] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *arXiv preprint arXiv:1606.00915*, 2016.

[28] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[29] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] A. A. Shvets, A. Rakhlin, A. A. Kalinin, and V. I. Iglovikov, "Automatic instrument segmentation in robot-assisted surgery using deep learning," in *2018*

17th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2018, pp. 624–628.

[31] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen, "Panoptic-deeplab," *arXiv preprint arXiv:1910.04751*, 2019.

[32] T. Chen, B. Xu, C. Zhang, and C. Guestrin, "Training deep nets with sublinear memory cost," *arXiv preprint arXiv:1604.06174*, 2016.

[33] Y. He, M. Keuper, B. Schiele, and M. Fritz, "Learning dilation factors for semantic segmentation of street scenes," in *German Conference on Pattern Recognition.* Springer, 2017, pp. 41–51.

[34] S. Xie, H. Zheng, C. Liu, and L. Lin, "Snas: stochastic neural architecture search," *arXiv preprint arXiv:1812.09926*, 2018.

[35] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," *arXiv preprint arXiv:1611.00712*, 2016.

[36] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[37] A. Veit and S. Belongie, "Convolutional networks with adaptive inference graphs," *arXiv preprint arXiv:1711.11503*, 2017.

[38] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.

[39] T. Ross, D. Zimmerer, A. Vemuri, F. Isensee, M. Wiesenfarth, S. Bodenstedt, F. Both, P. Kessler, M. Wagner, B. Müller *et al.*, “Exploiting the potential of unlabeled endoscopic video data with self-supervised learning,” *International journal of computer assisted radiology and surgery*, vol. 13, no. 6, pp. 925–933, 2018.

[40] C. d. C. Rocha, N. Padoy, and B. Rosa, “Self-supervised surgical tool segmentation using kinematic information,” *arXiv preprint arXiv:1902.04810*, 2019.

[41] F. Mahmood, D. Borders, R. Chen, G. N. McKay, K. J. Salimian, A. Baras, and N. J. Durr, “Deep adversarial training for multi-organ nuclei segmentation in histopathology images,” *IEEE transactions on medical imaging*, 2019.

[42] 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.

[43] 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.

[44] K. Greff, R. K. Srivastava, and J. Schmidhuber, “Highway and residual networks learn unrolled iterative estimation,” *arXiv preprint arXiv:1612.07771*, 2016.

[45] A. Levinshtein, C. Chang, E. Phung, I. Kezele, W. Guo, and P. Aarabi, “Real-time deep hair matting on mobile devices,” in *2018 15th Conference on Computer and Robot Vision (CRV).* IEEE, 2018, pp. 1–7.

[46] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott, “A perceptually motivated online benchmark for image matting,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 2009, pp. 1826–1833.

[47] 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.

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

[49] G. Csurka, D. Larlus, F. Perronnin, and F. Meylan, “What is a good evaluation measure for semantic segmentation?.” in *BMVC*, vol. 27. Citeseer, 2013, p. 2013.

[50] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[51] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[52] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[53] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[54] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[55] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 472–480.

[56] D. Li, J. Yang, K. Kreis, A. Torralba, and S. Fidler, "Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8300–8311.

[57] Y. Zhang, H. Ling, J. Gao, K. Yin, J.-F. Lafleche, A. Barriuso, A. Torralba, and S. Fidler, "Datasetgan: Efficient labeled data factory with minimal human effort," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 145–10 155.

[58] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8110–8119.

[59] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[60] Y. Viazovetskyi, V. Ivashkin, and E. Kashin, "Stylegan2 distillation for feed-forward image manipulation," in *European Conference on Computer Vision*. Springer, 2020, pp. 170–186.

[61] "Miccai 2015 endoscopic instrument segmentation and tracking dataset," http://endovissub-instrument.grand-challenge.org/, accessed: 2016-05-30.

[62] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[63] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution,

and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[64] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," *arXiv preprint arXiv:1611.08408*, 2016.

[65] A. Bansal, X. Chen, B. Russell, A. Gupta, and D. Ramanan, "Pixelnet: Representation of the pixels, by the pixels, and for the pixels," *arXiv preprint arXiv:1702.06506*, 2017.

[66] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention.* Springer, 2015, pp. 234–241.

[67] D. Pakhomov, W. Shen, and N. Navab, "Towards unsupervised learning for instrument segmentation in robotic surgery with cycle-consistent adversarial networks," *arXiv preprint arXiv:2007.04505*, 2020.

[68] Y. Babakhin, A. Sanakoyeu, and H. Kitamura, "Semi-supervised segmentation of salt bodies in seismic images using an ensemble of convolutional neural networks," in *German Conference on Pattern Recognition.* Springer, 2019, pp. 218–231.

[69] Y. Wu and K. He, "Group normalization," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[70] L. N. Smith, “Cyclical learning rates for training neural networks,” in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 464–472.

[71] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, “Averaging weights leads to wider optima and better generalization,” *arXiv preprint arXiv:1803.05407*, 2018.

Vita

Daniil Pakhomov received his BS in computer science from Saint Petersburg State University and his MS in computer science from Technical University of Munich. Next, he joined the PhD program at Johns Hopkins, where he has been advised by Nassir Navab. Daniil's current research focuses on unsupervised and data-efficient image segmentation for medical images and images of general domain. Daniil has published in diverse venues such as the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI) and International Conference on Intelligent Robots and Systems (IROS). Daniil won multiple prestigious machine learning competitions hosted by ECCV and CVPR computer vision conferences. During his PhD, he received an Intuitive Surgical Fellowship and an Excellence in Teaching Award.