Image-based segmentation and localization of surgical instruments using deep neural networks

Bachelor Thesis

Amelie Wagner

Division Translational Surgical Oncology
NCT Dresden

Reviewer:	Prof. Dr.-Ing. Stefanie Speidel
Second reviewer:	Dr.-Ing. Sebastian Bodenstedt
Advisor:	Isabel Funke

Duration: July 02, 2018 – November 03, 2018

NCT
NATIONAL CENTER
FOR TUMOR DISEASES
PARTNER SITE DRESDEN
UNIVERSITY CANCER CENTER UCC

TECHNISCHE
UNIVERSITÄT
DRESDEN

Ich versichere hiermit, die vorliegende Arbeit selbstständig angefertigt zu haben. Die verwendeten Hilfsmittel und Quellen sind im Literaturverzeichnis vollständig aufgeführt.

Dresden, den 03 November, 2018

Abstract

Segmentation and localization of surgical instruments in endoscopic videos during minimally invasive surgery is a fundamental requirement for supporting the surgeon during this sophisticated surgical technique. Here, a CNN-based deep learning method is used to get the positional information of the instrument directly out of the video frames without further modifications. In this work, the segmentation network TernausNet-11 is extended to a network that also solves the localization task. The instrument segmentation is addressed as a binary problem, where every pixel of the input image is labeled as instrument or background. The instrument location is obtained by the prediction of heatmaps where the center point of the instrument is located. That makes it possible for the network to use the information learned for the segmentation task to improve the localization results, as the weights of the network are shared for both tasks. All proposed methods are evaluated on the robotic dataset of MICCAI 2015 Endoscopic Vision subchallenge according to the challenge guidelines. The results of the experiments show that the proposed method is able to solve the segmentation task on the same level as state-of-the-art results. For the localization task, the models tested particularly on the test datasets achieve lower results than current state-of-the-art methods. The models trained in leave-one-surgery-out fashion outperform current state-of-the-art methods.

Contents

1. Introduction

1.1 Motivation

Minimally Invasive Surgery (MIS) has a considerable impact on modern surgical practice: By inserting specialized instruments (see figure 1.2) and a laparoscope through small access ports, the surgeon can operate on the internal anatomy under direct video observation of the surgical site without making large incisions (see figure 1.1). MIS performed in the abdomen and pelvis is also referred to as *laparoscopic surgery*. [1]

Because the surgical instruments required for this technique are smaller and the advanced instrument design improves tissue manipulation, the surgical trauma for the patient after surgery is reduced [4]. Rutherford et al. [5] showed in a laparoscopic adrenalectomy study that the postoperative inpatient stay was decreased from 9.8 to 5.1 days in comparison to the commonly known open surgery approach. Lacy et al. [6] showed in a study regarding colorectoral cancer operation that patients treated with MIS recovered faster, had shorter hospital stays, and higher probability of cancer-related survival.

Before it is possible to perform MIS, the surgeons and the rest of the clinical team have to be trained to operate with MIS instruments. There are several challenges compared to open surgical approaches [7]: The sense of touch for the surgeon is reduced and the view of the surgery is restricted by the endoscopic camera. It is also important to integrate image guidance to protect critical structures and help the surgeon to locate anatomical targets. Real-time knowledge of the position of the instruments within the surgical field of view is one of the main goals to improve MIS. [8]

The following approaches are currently used to overcome the instrument localization challenge in MIS: Electromagnetic tracking, which is realised by attaching an electromagnetic device to the instrument [9]. Optical localizers, where markers are applied to the surgical instrument and an additional external camera is used to recognize these markers and calculate the position of the instrument. This is the most widespread method in clinical use [10]. Other common methods are gaining positional information out of robotic surgery systems [3], ultrasound-based methods [11], and localization based on the endoscopic video used by the surgeon to operate [12].

1.2 Goals

Image-based localization approaches are highly attractive because they do not require modification of the instrument design or the operating room, like for example electromagnetic

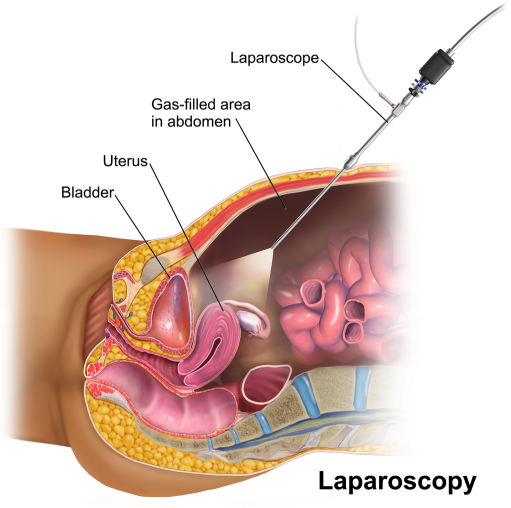


Figure 1.1: Illustration of MIS by Blausen et al. [2] The surgeon is getting the view of the operation scene by the laparoscope as real-time video.

(a) (b)

Figure 1.2: Examples of different MIS instruments: (a) Da Vinci [3] articulated robotic instrument, (b) rigid laparoscopic instrument [4].

tracking or optical localizers: The position and motion information is obtained directly from endoscopic camera used by the surgeon to operate. This is also the method with the lowest expenses. [4] The aim is therefore to develop an image-based localization method for assistance and improvement of MIS.

The input to the segmentation and localization methods proposed in this work are images out of an endoscopic video. The images contain surgical instruments. Segmentation means partitioning the input image into instrument part and background part. The partitioned image contains high pixel values where the instrument is located and low pixel values where the background is located. Localization means extracting the pixel position of specific instrument landmarks out of the input image. In this work, the center point of the shown instruments is located.

Convolutional Neural Networks currently provide the best solutions for image segmentation [13, 14]. This image segmentation task can be enhanced to localization of surgical instruments in images, as recently shown by Laina et al. [12]. In this work, the segmentation CNN proposed by Shvets et al. [15] is extended with the aim to solve the localization task based on previously learned segmentation information.

2. Background

Convolutional Neural Networks have several advantages in image recognition in comparison to more conventional computer vision techniques [14, 16]. They can be seen as an enhancement of Neural Networks.

2.1 Neural Network

A *Neural Network (NN)* (see figure 2.1) is an architecture that consists of a number of *neurons*. These are interconnected in different ways and often organized into layers [17].

One neuron consists of several *inputs* x_i, an *activation function* α that processes these inputs, and one *output* o.

$$o = \alpha\left(\sum_{i=1}^{n} x_i \cdot w_i + b_k\right) \tag{2.1}$$

Every input value of a neuron is scaled by its corresponding *weight* w_i. The *bias* b_k influences the behaviour of the neuron by increasing or decreasing the sum of the scaled input values for the calculation of the output of the neuron. The number of summands n denotes the number of input values of the neuron.

The specific output of the neuron depends on the kind of activation function α: Common activation functions are the sigmoid function (equation 2.2) and the rectified linear unit (ReLU) (equation 2.3). The input of the activation function ($\sum_{i=1}^{n} x_i \cdot w_i + b_k$) is abbreviated as z.

$$sigmoid(z) = \frac{1}{1 + e^{-z}} = \frac{e^z}{e^z + 1} \tag{2.2}$$

$$rec(z) = z^+ = \max(0, z) \tag{2.3}$$

For an example neuron structure with more detailed description see figure 2.2.

As the NN consists of many neurons, it consists also of the corresponding weights w_i and biases b_k. The weights and biases are also referred to as *parameters*. The neurons of a NN are arranged into *layers*. These layers are distinguished into one input layer, hidden layers

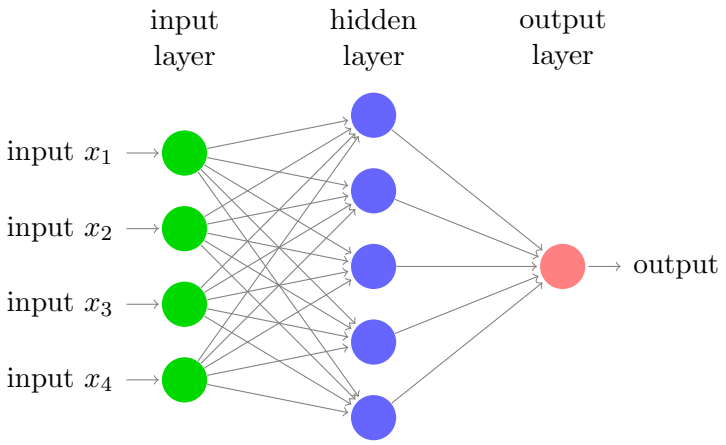


Figure 2.1: Example structure of a fully connected NN: Every circle denotes one neuron. This example network consists of three layers. The weights of the connections between the neurons are omitted in this figure.

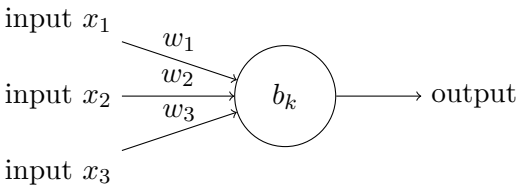


Figure 2.2: General structure of a *neuron*: The input values of the neuron are x_1, x_2, x_3. These input values are scaled by their corresponding weights w_1, w_2, w_3. b_k denotes the bias. In this example, the output of the neuron using the sigmoid activation function is calculated by $o = sigmoid\left(\sum_{i=1}^{3} x_i \cdot w_i + b\right)$.

and one output layer. In the network example in figure 2.1, the green layer is referred to as input layer, the red layer as output layer. The layers in between input and output layer, in the example one blue layer, are referred to as hidden layers. The network example is a *Fully Connected Network (FCN)*: The output of each neuron in one layer is connected to each neuron in the following layer [14].

In this work, the calculation of the final output of the network with respect to its current weights and biases is denoted as $y_{w,b}(x)$. x contains the input values to the network and can for example have vector or matrix shape.

2.1.1 Training Data

Training data is used for the training process (see section 2.1.2) of a NN. This training data is often composed of several datasets, later referred to as *training sets*. One training set $\{(x_1, t_1), ..., (x_j, t_j), ..., (x_N, t_N)\}$ contains N *training samples* (x_j, t_j) with one *training input* x_j and the corresponding *target* t_j.

The input given to the NN is denoted as x_j. The desired output of the network for the corresponding x_j is t_j. The target t_j is also referred to as *ground truth* or *label*. Since for each input x_j the corresponding ground truth t_j is known, this is referred to as a *supervised* machine learning problem.

When there is little training data available, the network can not be trained sufficiently to predict the desired output. The approach to train the network for many iterations with the same, small training dataset can cause overfitting (see section 2.1.3). To avoid this problem,

a commonly used method is *data augmentation*: The dataset is artificially enlarged by operations on the different training samples (x_j, t_j). Commonly used transformations are, given that the input of the network are images, translation [18], horizontal and vertical reflections, and altering the RGB channels of the images. [16]

2.1.2 Network Training

The general aim of training a NN is that, after the training is finished, the network recognizes certain features of the input data and classifies the input accordingly. These features are learned by each weight w_i and each bias b_k of the network during the training phase. The resulting network out of a network structure that learned specific parameters is referred to as *model*.

The prediction result of a NN with given input x is calculated as $y_{w,b}(x)$. During the training period, the deviation between $y_{w,b}(x_j)$ for the training input x_j and the corresponding desired output t_j is calculated and referred to as error E_j. The aim is to adjust the parameters of the network in a way that E_j is minimized.

The *loss function* $L(y_{w,b}(x_j), t_j)$ of the NN specifies the way how E_j is determined for one training sample (x_j, t_j). The loss function is also often referred to as cost function.

One of the most common loss functions is the mean squared error (MSE):

$$MSE(y_{w,b}(x_j), t_j) = (t_j - y_{w,b}(x_j))^2 \tag{2.4}$$

As the loss $L(x_j, t_j)$ is calculated for each training sample , the results are summed and then averaged over the complete training set $\{(x_1, t_1), ..., (x_j, t_j), ..., (x_N, t_N)\}$. This calculation of the overall error E is stated in equation 2.5 for N training inputs.

$$E = \frac{1}{N} \sum_{j=1}^{N} L(y_{w,b}(x_j), t_j) \tag{2.5}$$

E is reduced by adjusting the weights and biases of the network with the *gradient descent* method [19]. To reduce E, the partial derivatives of the loss function with respect to all G weights w_i and all D biases b_k of the network are calculated stepwise. Afterwards, each weight w_i and each bias b_k in $y_{w,b}(x_j)$ is adjusted. The partial derivatives are contained in two *gradients* of the loss function: ∇E_w for the weights and ∇E_b for the biases of the network.

$$\nabla E_w = \left(\frac{\partial E}{\partial w_1}, ..., \frac{\partial E}{\partial w_i}, ..., \frac{\partial E}{\partial w_G} \right)^T \tag{2.6}$$

$$\nabla E_b = \left(\frac{\partial E}{\partial b_1}, ..., \frac{\partial E}{\partial b_k}, ..., \frac{\partial E}{\partial b_D} \right)^T \tag{2.7}$$

Every weight w_i and every bias b_k in the loss function is updated to w_i' and b_k' by using the entries of ∇E_w and ∇E_b.

$$w_i' = w_i - \lambda \frac{\partial E}{\partial w_i} \tag{2.8}$$

$$b_k' = b_k - \lambda \frac{\partial E}{\partial b_k} \tag{2.9}$$

The parameter λ refers to the *learning rate*. It is used to control the learning process of the network by scaling the adjustment of the weights and biases.

The gradients ∇E_w with G entries and ∇E_b with D entries are summarized into one gradient ∇E.

$$\nabla E = \left(\frac{\partial E}{\partial w_1}, ..., \frac{\partial E}{\partial w_i}, ..., \frac{\partial E}{\partial w_G}, \frac{\partial E}{\partial b_1}, ..., \frac{\partial E}{\partial b_k}, ..., \frac{\partial E}{\partial b_D} \right)^T \tag{2.10}$$

Each entry $\frac{\partial E}{\partial w_i}$ and $\frac{\partial E}{\partial b_k}$ of ∇E is calculated separately for every training sample (x_j, t_j), and then averaged over all N training samples.

$$\nabla E = \frac{1}{N} \sum_{j=1}^{N} \nabla E_j \tag{2.11}$$

Calculating the gradient over all the training samples and afterwards averaging the result is computationally expensive. To reduce the learning time of the network, it is common to use a modified gradient descent method: It is denoted as *stochastic gradient descent*. The gradient ∇E is estimated by calculating the gradient for a subset of β samples out of the training set, and then averaging the result: $\frac{1}{\beta} \sum_{i=1}^{\beta} \nabla E_i$. The subset of the original training set is referred to as *mini-batch*. The number of elements of the mini-batch, β, is the *batch size*. Training the NN with randomly picked mini-batches also has the advantage of avoiding shifts of the network. [20] When the complete training dataset was shown to the network, i.e. every element of the training dataset was propagated through the network one time, one *epoch* of training is completed.

Backpropagation (BP) [21] is one way to efficiently calculate all partial derivatives of the network [19]. It is by far the most common approach for training NNs [22]. First, the forward pass of the training phase is performed: The output of the NN is calculated out of the given input. The gradients ∇E_j are calculated for each input of the mini-batch by backpropagating the error $E_j = L(y_{w,b}(x_j), t_j)$ back through the network. Then, the update step is performed: Every weight w_i and bias b_k of the network is corrected according to the calculated gradients. [21] [23]

2.1.3 Network Testing

When training a NN, the aim is to obtain a network with optimal generalization performance. Generalization performance means that E is minimal on network inputs not seen during training. This E is also referred to as *generalization error*. [24]

Therefore, when testing the performance of a NN, it is necessary to provide input that was not given to the network during the training procedure. These previously unseen input data is referred to as *test data*. This test data is composed of several *test sets*. One test set $\{(\mathbf{x_1}, \mathbf{t_1}), ..., (\mathbf{x_l}, \mathbf{t_l}), ..., (\mathbf{x_M}, \mathbf{t_M})\}$ consists of M *test samples* $(\mathbf{x_l}, \mathbf{t_l})$ with one test input $\mathbf{x_l}$ and its corresponding target $\mathbf{t_l}$.

When the generalization error increases during training, this indicates a problem that is denoted as *overfitting*: While the network seems to get better and better during training, i.e. E on the training dataset decreases, the network actually begins to predict worse on the test set, the generalization error increases. [24] The prediction performance of the network is fitted to the training dataset, but generalizes poorly.

It is a commonly used method to split the given data into separated training and testing sets and evaluate the network with the *cross-validation* method.

In *k-fold cross-validation*, all training data is assigned randomly to k datasets with approximately the same size. These k datasets are separated into $k-1$ training sets and in one remaining test set. The NN is trained on the $k-1$ training sets and then evaluated on the remaining test set. This procedure is done k times. Each time another set is chosen as test set, consequently the training sets are also different each time. As a result, there are k different models of the NN. The performance of the NN is evaluated by testing each model on its corresponding test set and averaging the results over all models.

Another variant of k-fold cross-validation is *leave-one-out cross-validation (LOOCV)*: For the given training data with N elements, one sample $(\mathbf{x_l}, \mathbf{t_l})$ is chosen as test sample, the rest of the data as training samples. Every element is chosen once as a test sample. The performance of the resulting models is evaluated the same way as for k-fold cross-validation, with $k = N$.

Cross-validation is a useful method to estimate the generalization error: The evaluation of the network includes all available labeled data without having used it for training and testing at the same time. [25]

A NN structure which has already adjusted weights and biases due to previous training on several datasets, for example ImageNet [26], is also referred to as *pre-trained* network. Using pre-trained networks instead of randomly initialized networks to solve tasks such as image recognition reduces both E and overfitting, especially when there are few training examples available[16, 26].

2.2 Convolutional Neural Network

A *Convolutional Neural Network (CNN)* is a specific kind of NN. The first application of CNNs was developed by LeCun et al. [27]. In the following, the structure of a CNN is explained, given that it is used for image recognition.

In this scenario, the input to a CNN consists of an image with three dimensions. For example a RGB image has two dimensions for the resolution and one dimension for the RGB channels. This input is referred to as *input layer*, the dimensions as *channels*.

One CNN consists of multiple *feature maps*, see for example U-Net in figure 2.3. These feature maps usually have three dimensions: Two spatial dimensions for width and height, and a third dimension which denotes the number of channels. The input image is also a feature map.

Convolutions with a specific kernel size are applied to the feature maps, followed by an activation function α (see section 2.1). The result of the convolutions is stored in the following feature map of the CNN. The kernel is also referred to as *filter*. In a CNN structure, the parameters of the neurons are not realized with direct connections between neurons like in a general NN: They are stored in the filters of the CNN. The learning process with gradient descent works like for NNs in general, but in particular the weights of the filters of the CNN are updated.

The number of feature map dimensions and filter dimensions is always the same. The first two dimensions of the filter are applied to the spatial dimensions of the feature map. These dimensions of the filter are usually smaller than the corresponding feature map dimensions it is applied to. The third dimension of the filter always has the same size as the number of channels of its input feature map. This means that the filter extends through the full depth of its input volume. [28] The *stride* of a filter denotes the distance between the regions of the feature map it is applied to.

The state-of-the-art CNNs, especially for image segmentation, consist of one *downsampling phase* followed by an *upsampling phase*. These are also denoted as *encoder* and *decoder* of

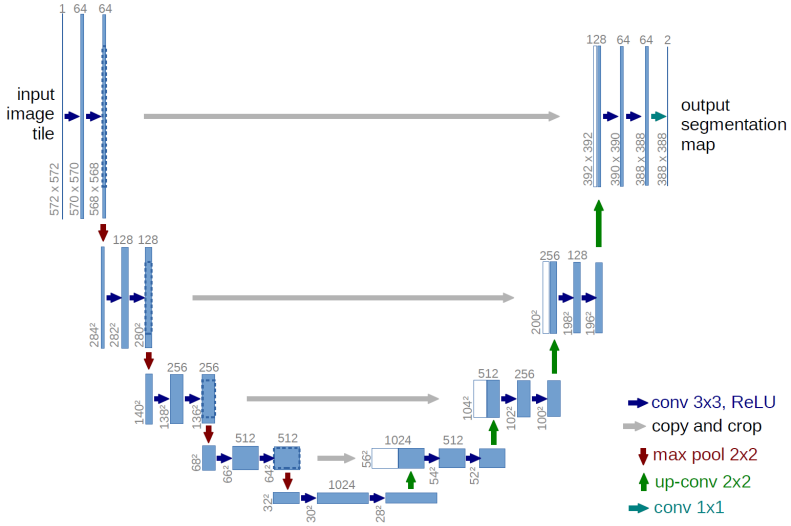


Figure 2.3: Structure of U-Net, proposed by Ronneberger et al. [30]. Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top, the resolution of the map is denoted at the left side of the box. White boxes refer to feature maps that are copied from the encoder phase and concatenated to the decoder phase. The arrows denote the operation types of the network.

the network. In the downsampling phase, the input of the network is reduced stepwise by convolutions. In the upsampling phase, the size of the output after the downsampling phase is increased again by application of deconvolutions. [15, 29] U-Net (see figure 2.3) proposed by Ronneberger et al. [30] is a typical encoder-decoder CNN used for segmentation of images (see figure 2.3).

There exist different kinds of convolutions. In the encoder phase of the network, the applied convolutions mostly preserve or reduce the spatial dimensions of the input and increase the number of channels. In the decoder phase, the convolutions mostly preserve or increase the spatial dimensions of the input and decrease the number of channels. This type of convolution is referred to as *transposed convolution*, *deconvolution* or *upconvolution*. To increase the spatial dimensions, *padding* is included for the convolutions.

Pooling in CNNs is realized by a filter that reduces the dimension of its input. It is a widely used method to downsample the feature maps in the encoder phase of the network. The most common pooling method is *max pooling*: Only the maximum value of the input region of the filter is returned.

3. State of the Art

The following works propose current state-of-the-art solutions for surgical instrument segmentation and localization in endoscopic videos.

Bodenstedt et al. [13] present the results of the methods submitted for EndoVis15 (see section 5.1.2). Additionally, they propose merged results from the submitted models because it significantly increases accuracy in comparison to the best submitted stand-alone method. These results show that the CNN-based methods reached the highest performance compared to other methods. Therefore, in the following part of this chapter, only CNN-based methods concerning image-based segmentation of surgical instruments will be described. For EndoVis15-S, SEG-KIT-CNN is proposed: It is a FCN that uses VGG-16 [31] as an encoder.

Garcia et al. [32] propose two adapted FCNs for EndoVis15-S: FCN and FCN-real-time. The forward evaluation takes 100ms for the FCN which is below the frame rate of an endoscopic video. The frame rate of an endoscopic video is usually 25 frames per second (fps), 30 fps or 60 fps. The FCN-real-time approach overcomes this problem by estimating an affine transformation on the instrument between the time of two segmentations of the network. When a frame is read from the surgical video feed, it is sent to the FCN-real-time only if the network is not busy processing the previous frame.

Pakhmov et al. [33] propose a ResNet-101 which is converted into a FCN for EndoVis15-S (ResNet-FCN).

Attia et al. [34] provide a Convolutional Neural Network - Recurrent Neural Network (hybrid CNN-RNN) auto encoder-decoder network (see figure 3.1) for instrument segmentation for EndoVis15-S.

Laina et al. [12] propose three different CNN-based approaches to predict the location of surgical instruments (see figure 3.2). They are abbreviated as L-network, SL-network and CSL-network. All of the proposed networks share ResNet-50 [35] as encoder part.

The decoder part of the L-network regresses the positions of the instrument directly as a $2\text{x}n_i$ dimensional vector, where n_i is one predicted location of one instrument. This method is frequently used in literature. [36] The segmentation task of the instrument is excluded in this approach.

In the SL-network, the prediction of the 2D locations of the instrument and the segmentation of the instrument are combined in one architecture. Both tasks share weights along the

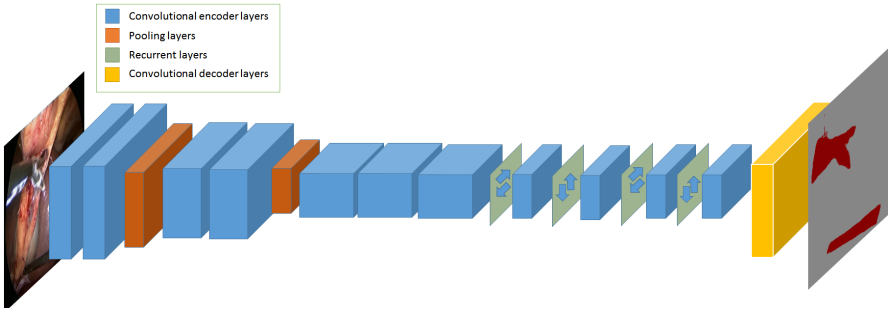


Figure 3.1: Proposed hybrid CNN-RNN architecture by Attia et al. [34]. The encoder phase of the network consists of seven convolutional layers denoted as blue blocks, the two max-pooling layers are denoted as orange blocks. The feature maps extracted by the encoder part of the network are fed into 4 layers of the recurrent network, denoted as green masks with 4 decoupled directions. The output mask of the instrument is reconstructed using an auto decoder network, denoted with yellow blocks.

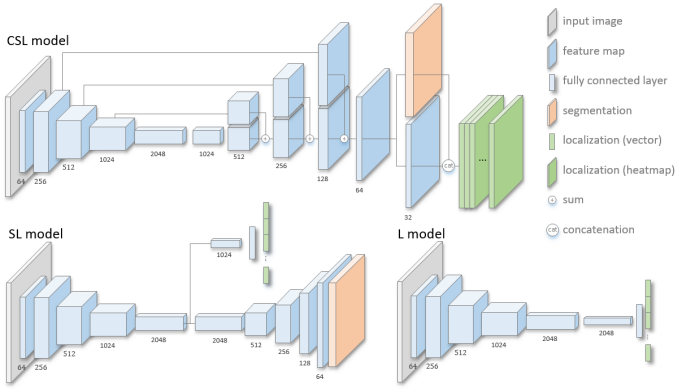


Figure 3.2: The three proposed network architektures in Laina et al. [12]

encoder part and split into two distinct parts at the decoder part of the network. The prediction of the instrument positions is solved like in the L-model. For the segmentation task, the probability of each pixel of belonging to the instrument or the background is predicted.

The CSL-network predicts the position of the instrument by regressing a heatmap instead of the exact coordinates. The heatmap for training is created by applying a two-dimensional Gaussian function to the ground truth position of the instrument (see section 4.3). By applying this method, the problem that ground truth annotations of the instrument position can differ by several pixels is avoided. The segmentation and localization task share weights over the entire network because the heatmaps have the same size as the segmentation masks. The CSL-network outperforms the two other proposed methods.

Du et al. [29] solved the localization task by using a fully convolutional detection-regression network (FCN-det-reg) (see figure 3.3), where the instrument joints and associations between joint pairs are located by the detection subnetwork and subsequently redefined by the regression subnetwork.

Positions of the surgical instruments are inferred using maximum bipartite graph matching [37]. The EndoVis15-T (see section 5.1.2) ground truth annotations were modified and replaced by self-generated annotations (see figure 3.4).

The detection part of the network (see figure 3.3) consists of two branches. The first

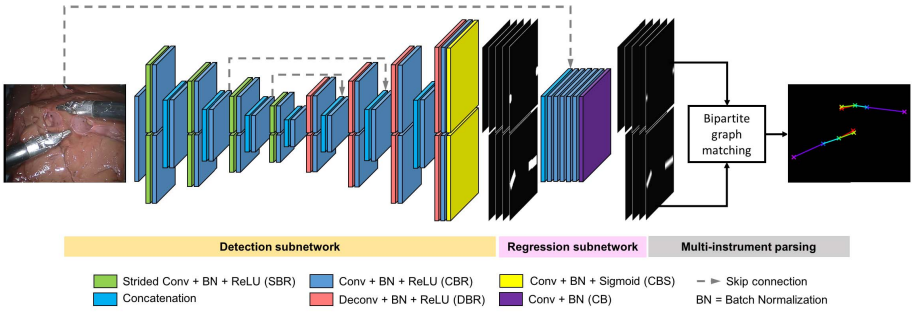


Figure 3.3: Structure of the detection-regression network proposed in Du et al. [29]

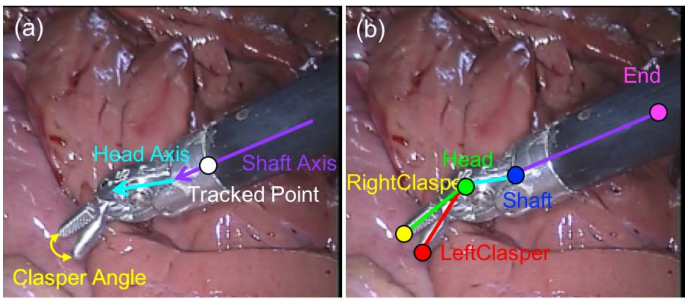


Figure 3.4: The original (a) and modified (b) ground truth annotations for EndoVis15-T.

branch is used to predict probability maps for each joint of the surgical instrument, the second branch is used to predict association probability maps for each joint pair. From the prediction of the detection part, a coarse position of the instrument is obtained. With the regression part of the network, that takes the prediction of the detection part as an input, refined positions of the joints are obtained. The ground truth for the regression subnetwork for each instrument joint location consists of heatmaps formed by a two-dimensional Gaussian distribution at the labeled position of the joint. The ground truth for the association between joint pairs are formed by a Gaussian distribution along the joint pair center line. The approach to use heatmaps in form of images to predict the instrument position is similar to Laina et al. [12].

Notably, it is not possible to compare the FCN-det-reg results directly to the other results: The provided ground truth is modified, the results that exceed a certain threshold are excluded from evaluation, and the training procedure is different from the challenge guidelines.

Shvets et al. [15] used four different networks for EndoVis17-S (see section5.1.1): U-Net (see figure 2.3), TernausNet-11 and TernausNet-16 which are a modifications of U-Net, and one network that is a modification of LinkNet [38]. TernausNet-16 (see also figure 3.5) outperforms the other networks in the segmentation tasks except for the inference time. VGG16 [31], pre-trained (see section 2.1.3) on ImageNet, is used as encoder for TernausNet-16.

To solve specifically the localization task of EndoVis15-T, Bodenstedt et al. [13] propose two approaches that are not CNN-like: TRA-UGA [39] and TRA-KIT-RF [40].

An overview of the previously proposed methods is is given in table 3.1

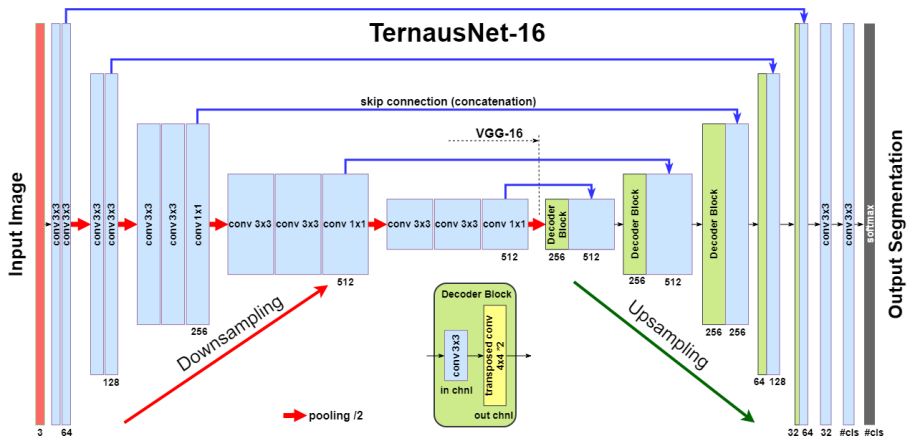


Figure 3.5: Structure of the segmentation network TernausNet-16. Each box corresponds to a multi-channel feature map with the number of channels denoted below the box. The height of each box represents the corresponding feature map resolution. The blue arrows illustrate skip-connections where information is transmitted from the decoder to the encoder phase of the network. [15]

EndoVis15-S results							
	b.acc.	accuracy	recall	prec.	spec.	Dice	IoU
CSL-network [12]	92.6	-	86.2	-	99.0	88.9	-
SEG-KIT-CNN [40]	-	96	77	86	-	81	-
SEG-UGA [39]	-	96	66	95	-	78	-
FCN [32]	83.7	-	72.2	-	95.2	-	-
FCN-real-time [32]	88.3	-	87.8	-	88.7	-	-
ResNet-FCN [33]	92.3	-	85.7	-	98.8	-	-
CNN-RNN [34]	93.3	-	90.4	-	96.1	-	82.7

Table 3.1: Results of EndoVis15-S (see section 5.1.2) for every proposed paper. Balanced accuracy (b.acc.), accuracy, recall, precision (prec.), specificity (spec.), Dice, Intersection over Union (IoU) are given in %. For definition of evaluation metrics see section 4.5.

EndoVis15-T results			
	distance LOSO	distance T4	distance-total
CSL-network [12]	20.68	49.68	34.46
TRA-KIT [40]	-	-	106.6
TRA-UGA [39]	-	-	34.9
FCN-det-reg [29] train set *thresh=20px*	-	-	3.36
FCN-det-reg [29] test set *thresh=20px*	-	6.96	-
FCN-det-reg [29] test set *thresh=30px*	-	7.98	-

Table 3.2: Results of EndoVis15-T (see section 5.1.2) for every proposed localization solution. The distance between predicted position of the instrument and labeled ground truth position of the instrument is given in pixels. The evaluations refer to the results for models trained according to LOSO fashion (distance LOSO), the results for models trained with all training sets and tested on the test sets (T4), and to the mean value of the results of both training methods (distance in total).

4. Methods

The goal in this work was to develop a CNN architecture that is able to predict surgical instrument segmentation and location in endoscopic images. All networks are evaluated on EndoVis15 (see section 5.1.2) and EndoVis17-S (see section 5.1.1) datasets. When two instruments are visible at the same time in the input video frame, they are not distinguished by the network. This is done in a postprocessing step.

The approach in this work is to use the information learned by a CNN for instrument segmentation to improve the instrument localization task. When transforming the localization ground truth to the same shape as the segmentation ground truth (see section 4.3), both tasks can share the parameters of the CNN. This seems to be an advantage in comparison to other localization approaches, as showed recently by Laina et al. [12].

4.1 Instrument Segmentation Network

In this work, TernausNet-11 is chosen as basis segmentation network and then extended to solve the localization task. TernausNet-11 (see figure 4.2) is proposed in the winning paper of EndoVis17-S by Shvets et al. [15]. The encoder part is a VGG11-Network [31] pretrained on ImageNet [16]. Every convolutional layer of the network is followed by a ReLU activation function.

The output of TernausNet-11 is a binary image, that contains a white segmentation mask where the instrument is located (see figure 4.1b).

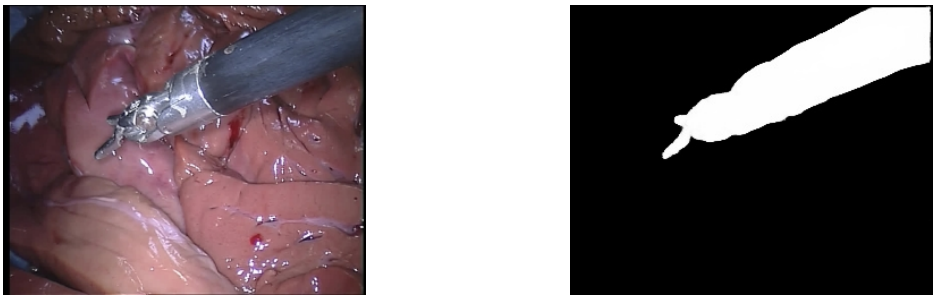
As TernausNet-11 outperforms the other networks for the different segmentation challenges except TernausNet-16, which has a larger inference time, TernausNet-11 is chosen as basis for the Localization Network (see section 4.2).

TernausNet-11 is trained for the image segmentation task using the loss function $L_{segm}(y_{w,b}(x_j), t_j)$. It consists of two parts:

$$L_{segm}(y_{w,b}(x_j), t_j) = BCE(y_{w,b}(x_j), t_j) - \log J_{binary}(y_{w,b}(x_j), t_j) \tag{4.1}$$

The first part is a binary cross entropy loss (BCE) combined with a sigmoid function:

$$BCE_{sigm}(y_{w,b}(x_j), t_j) = -\frac{1}{N}\sum_{i=1}^{N}[t_j \log(\sigma(y_{w,b}(x_j))) + (1 - t_j)\log(1 - \sigma(y_{w,b}(x_j))] \tag{4.2}$$



(a) Example input image.

(b) Predicted instrument segmentation frame by TernausNet-11.

Figure 4.1: Example input image out of EndoVis15-S with corresponding predicted mask by TernausNet-11.

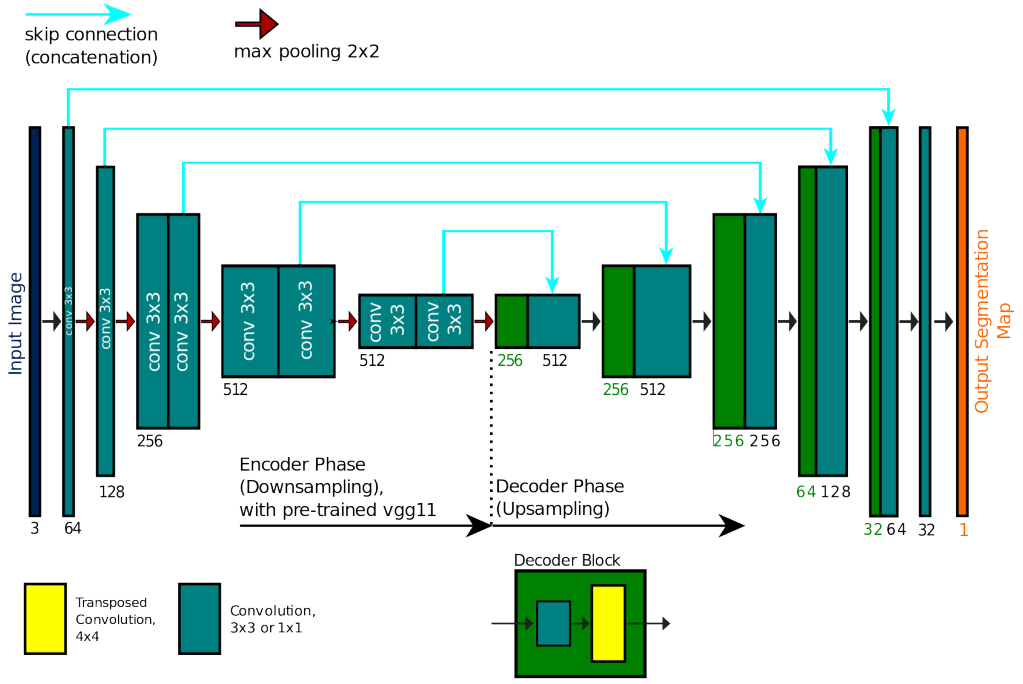


Figure 4.2: Structure of TernausNet-11. The network is proposed by Shvets et al. [15]. The blue boxes represent feature maps, the number of channels is denoted below each box. In the decoder phase of the network, each green box is a decoder block that consists of a convolution followed by a transposed convolution. The blue arrows denote skip connections: The feature maps from the encoder phase are concatenated to the decoder blocks in the upsampling phase of the network. The red arrows represent 2x2 max pooling operations. The orange box represents the segmentation output of the network.

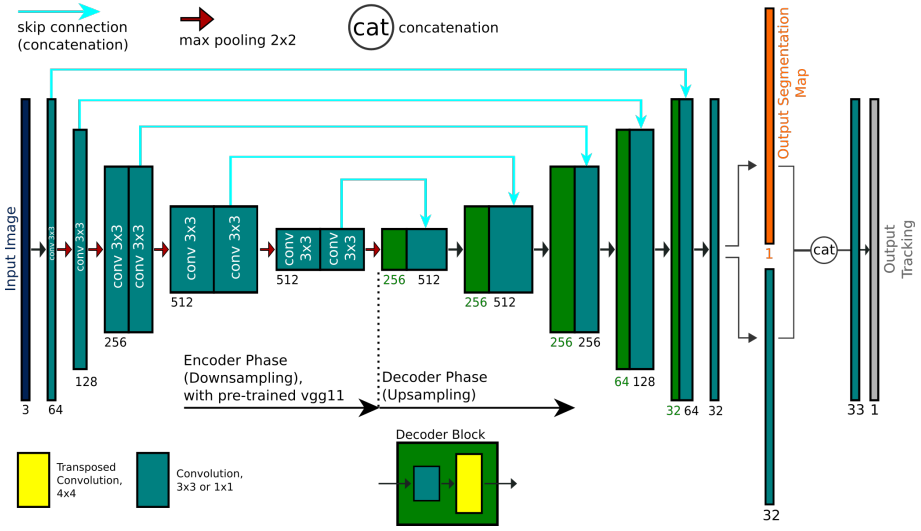


Figure 4.3: Network architecture that is used for segmentation and localization of the surgical instruments in this work. It is an extended TernausNet-11. The orange box represents the segmentation output, the grey box the localization output of the network.

Combining the sigmoid function with the BCE loss - instead of applying the sigmoid function first and BCE in a second step - increases the numerical stability. [41]

The second part of $L_{segm}(y_{w,b}(x_j), t_j)$ is a Jaccard loss for binary evaluation:

$$J_{binary}(y_{w,b}(x_j), t_j) = \frac{1}{N} \sum_{i=1}^{N} \frac{t_j \cdot y_{w,b}(x_j)}{t_j + y_{w,b}(x_j) - t_j \cdot y_{w,b}(x_j)} \tag{4.3}$$

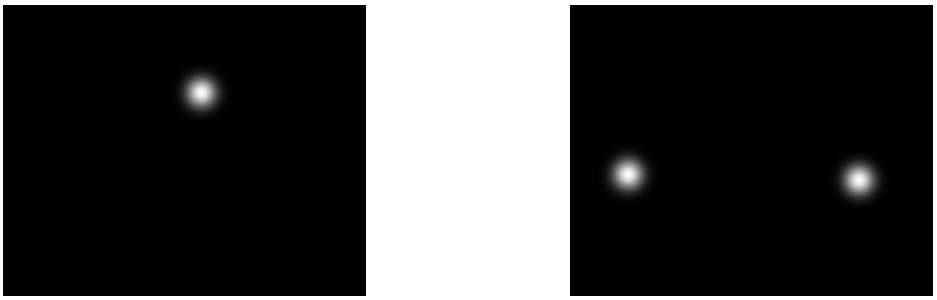
J_{binary} is a differentiable generalization of the Jaccard index that is adjusted to fit into the loss function of the network. It is a differentiable function and therefore suitable for using it for optimizing the network with gradient descent. [42]

4.2 Instrument Localization Network

The network architecture that is used for localization of surgical instruments in this work is a modified TernausNet-11 (see figure 4.3). It is extended by three more layers: One branching layer with 32 channels, one concatenation layer where the segmentation output and the branching layer are concatenated, and one output layer for localization of the instruments. This architecture is inspired by Laina et al. [12]. The network predicts greyscale images, therefore the segmentation output layer and the localization output layer each have one channel. The Localization Network is abbreviated as *LocNet*.

The output of the segmentation output layer is the same as the output of TernausNet-11. The output for the localization task is a greyscale heatmap (see section 4.3). A high pixel value at the predicted heatmap indicates a high probability that the instrument center point is located at this pixel position.

The output of LocNet is calculated with given input x_j as a tuple $(y_{w,b}(x_j), z_{w,b}(x_j))$, where $y_{w,b}(x_j)$ is the output of the segmentation layer and $z_{w,b}(x_j)$ is the output of the localization layer. Given the target s_j as ground truth for segmentation and t_j as ground truth for the localization heatmap, a combined loss for concurrent instrument segmentation and localization is defined as follows:



(a) Example heatmap generated from the ground truth position of one instrument.

(b) Example of a heatmap frame generated for the case of two visible instruments in the corresponding input image.

Figure 4.4: Example heatmaps generated out of the ground truth for EndoVis15-T.

$$L_{loc}(y_{w,b}(x_j), z_{w,b}(x_j), s_j, t_j) := (1-\gamma)L_{segm}(y_{w,b}(x_j), s_j) + \gamma MSE(z_{w,b}(x_j), t_j) \quad (4.4)$$

The segmentation loss is calculated by $L_{segm}(y_{w,b}(x_j), s_j)$, the localization loss is calculated by $MSE(z_{w,b}(x_j), t_j)$. One training sample for segmentation is denoted as (x_j, s_j), one training sample for localization as (x_j, t_j). The impact of each loss on the overall loss is scaled by the factor γ.

4.3 Heatmaps

It is necessary to convert the localization targets, given 2D image positions for EndoVis15-T, into greyscale images, in order to make it possible for the LocNet to process them. Out of each localization target one heatmap is created. The *heatmap* is a greyscale image that has higher pixel values around the center point position of the surgical instrument (see section 5.2).

The heatmaps are generated by calculating a two-dimensional Gaussian distribution $gauss2D(x_r, y_r)$, centered at the center point (x_{cp}, y_{cp}) of the instrument.

$$H(x_r, y_r) = gauss2D(x_r, y_r) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_r - x_{cp})^2 + (y_r - y_{cp})^2}{2\sigma^2}} \quad (4.5)$$

The standard deviation σ controls the spread of the Gaussian around the instrument location (x_{cp}, y_{cp}). σ^2 denotes the variance. For better interpretability, the pixel values are scaled to the range $[0, 1]$ before given to the network for training.

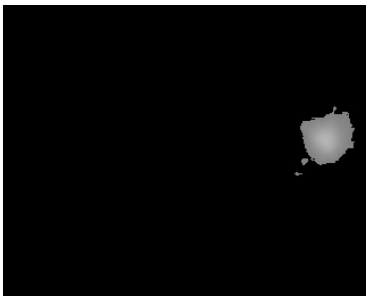
By iterating over the dimensions of the training input image and applying $gauss2D(x_r, y_r)$, a heatmap with the same dimensions is generated out of the ground truth instrument position. (x_r, y_r) is one pixel position of the training image, which corresponds to the pixel position in the generated heatmap.

When two instruments are visible in the input image, the heatmap is generated by calculating two Gaussian distributions, one for each instrument (see figure 4.4b).

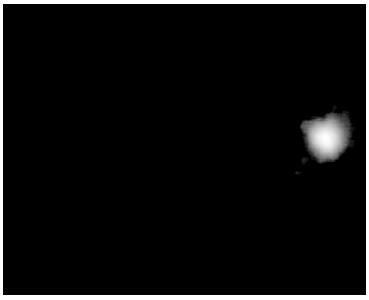
4.4 Postprocessing

The heatmaps predicted by the network are mostly having a pixel range from $[127, 255]$. In order to improve the extraction of the instrument position, the pixel values of the predicted heatmaps are thresholded with *threshold* set to 129 (see figure 4.5b):

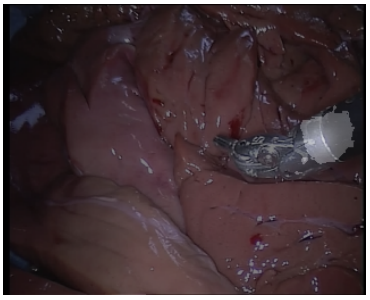
(a) Unmodified localization prediction of LocNet.



(b) Thresholded localization prediction of LocNet.



(c) Equalized localization prediction of LocNet in order to increase contrast.



(d) Input frame out of EndoVis-15 with superimposed postprocessed heatmap.

Figure 4.5: Example sequence for postprocessing a predicted heatmap. First, the original heatmap (a) is thresholded (b). Afterwards, the contrast is increased by histogram equalization (c). Figure (d) shows the postprocessed heatmap with underlying corresponding input image

$$thresh(H(x_r, y_r)) = \begin{cases} H(x_r, y_r) & \text{if } H(x_r, y_r) \geq threshold \\ 0 & \text{otherwise} \end{cases}$$

In a second postprocessing step, histogram equalization [43] is used to increase the contrast of the Gaussian distribution in the image (see figure 4.5c).

To get the instrument position out of the thresholded heatmap, *k-means* clustering is used. It is a commonly used method to automatically partition a dataset into k groups. [44] When the input to the corresponding predicted heatmap contains one instrument, k is set to 1. When two instruments are visible in the input image, k is set to 2. The pixel values of the thresholded heatmap are used as weights for the k-means clustering to improve the results, because a higher pixel value indicates a higher probability that the instrument center is located at that position.

4.5 Evaluation Metrics

The following section gives an overview of the evaluation metrics that are commonly used for the validation of NNs. As the network in this work is used for binary classification, the evaluation methods proposed in this section cover only the evaluation of predictions with two classes. The descriptions are adjusted for evaluating LocNet. There are two classes for binary classification: *Positive* if a pixel in an input image is part of a surgical instrument and *negative* if the pixel belongs to the background. The prediction of a model for one pixel is evaluated as *true* if the predicted class is correct, or *false* if it is wrong.

Each output predicted by a model can be classified in one of four categories:

- TP (*true positive*)
  Represents the number of pixels correctly labeled as surgical instrument.
- TN (*true negative*)
  Represents the number of pixels correctly labeled as background.
- FP ($false\,positive$)
  Represents the number of pixels which are erroneously labeled as surgical instrument but actually are part of the background.
- FN ($false\,negative$)
  Represents the number of pixels which are erroneously labeled as background but actually are part of a surgical instrument in the image.

The *precision* evaluates the proportion of predicted positive classes ($TP + FP$) that are actually positive classes (TP).

$$\text{precision} = \frac{TP}{TP + FP} \tag{4.6}$$

The evaluation metric *recall* (see equation 4.7), in literature also denoted as *sensitivity*, *hit rate* or *true positive rate*, is the proportion of actual positive classes ($TP + FN$) that are correctly predicted as positive (TP). [45]

$$\text{recall} = \frac{TP}{TP + FN} \tag{4.7}$$

The evaluation metric *specificity*, also called *true negative rate*, measures the proportion of all pixels that correspond to the negative class ($TN + FP$) that are correctly identified as this class (TN). It can be seen as inverse recall because it works the same way as the recall metric, but takes into account the negative examples instead. [45]

$$\text{specificity} = \frac{TN}{TN + FP} \tag{4.8}$$

The evaluation metrics precision and recall focus only on the positive examples and predictions. Neither of them captures information about how the model is handling negative cases. [45] *Intersection over Union (IoU)*, also referred to as *Jaccard index*, can be seen as a combination of these two metrics. It is considered as a better segmentation metric for evaluating binary images because it penalizes both over- and undersegmentation. [34]

$$\text{IoU} = \frac{TP}{TP + FP + FN} \tag{4.9}$$

Dice (see equation 4.10) measures the overlap between two binary images [46]. It is similar to IoU, which counts TP only one time in both the numerator and denominator.

$$\text{dice} = \frac{2TP}{2TP + FP + FN} \tag{4.10}$$

Accuracy is the proportion of all correct predictions ($TP + TN$) among the total number of cases ($TP + TN + FP + FN$). The advantage over recall, precision, and IoU is that accuracy explicitly takes into account the classification of negative cases [45].

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.11}$$

The appropriateness of evaluating a model with the accuracy metric depends on the distribution of the dataset. High accuracy results can depend on the distribution of the dataset: Assumed in a surgical instrument dataset the instrument is very small or there are many sequences where no instrument is visible, a model that always predicts the negative case that only background is visible, will achieve high accuracy because the FN rate will be very low. [47] Using *balanced accuracy (b.acc.)* for the evaluation decreases this dependency on the dataset. [48]

$$\text{balanced accuracy} = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{4.12}$$

5. Evaluation

5.1 Datasets

In this work, two datasets for segmentation and one for localization of surgical instruments are used for training and testing of the networks.

5.1.1 Endoscopic Vision Challenge 2017

One segmentation dataset used in this work is part of the subchallenge Robotic Instrument Segmentation of the MICCAI Endoscopic Vision Challenge 2017. The subchallenge is divided into the tasks binary segmentation, part recognition, and type distinction of surgical instruments. Binary segmentation means that the instruments are distinguished from the background. [49] This challenge is abbreviated as *EndoVis17-S*.

Robotic Training Data

The eight training sets of EndoVis17-S consist of stereo camera images, each with a resolution of 1280×1024 px (width×height) acquired from a Da Vinci Xi robot [50] during several different porcine procedures (see figure 5.1b). The ground truth for the camera images are hand annotated 1280×1024 px greyscale images. It is provided for the part and type distinction task. These greyscale images are referred to as *masks*. For the part recognition task, each instrument part is encoded with different numerical values (see figure 5.1a).

(a) Example ground truth with increased contrast and brightness for better visibility.

(b) Example image with three instruments visible.

Figure 5.1: Unmodified example image and corresponding ground truth for one instrument out of EndoVis17-S.

For the instrument type distinction task, the ground truth for each instrument is provided separately: When more than one instrument is visible in the input image, each instrument has one ground truth image (see figure 5.1).

Robotic Test Data

The test data consists of eight datasets with 75 images each. They are acquired the same way as the training data. The test sequences were sampled immediately after each training sequence. Additionally, there are two test sets with 300 images each. They contain different procedures than the eight datasets proposed for training.

5.1.2 Endoscopic Vision Challenge 2015

Parts of the data used for this work are proposed in the subchallenge Instrument Segmentation and Tracking of the MICCAI Endoscopic Vision Challenge 2015 [51]. The subchallenge in general is abbreviated as *EndoVis15*.

This subchallenge is separated into two dataset types: One *rigid set* that contains only rigid surgical instruments and one *robotic set* that contains only robotic surgical instruments. The rigid set contains laparoscopic instruments during colorectal surgeries. It reflects typical challenges in endoscopic vision like occlusion, smoke and bleeding. In the robotic set, the instruments show typical poses and articulation in robotic surgery. There is some occlusion, but no smoke and bleeding in any sequence.

This work only uses the robotic set. It is separated in a segmentation and a localization part. [51] The training and test inputs are the same, the ground truth is different for each part. The segmentation part is abbreviated as *EndoVis15-S*, the tracking part is abbreviated as *EndoVis15-T*.

Robotic Training Data

There are four robotic training sets. Each consists of a different 720×576 px ex-vivo surgery video with 1100 frames per video and corresponding ground truth for each frame. *Ex-vivo* refers to procedures that are realized in an artificial environment outside a living organism. The four surgeries are performed using the Da Vinci Surgical System [52].

The first training set displays two instruments, the remaining three sets contain one instrument.

For EndoVis-S, the ground truth is provided as a greyscale video. For every frame, the (R,G,B) values of each pixel are labeled as background (0,0,0), shaft (160,160,160) or manipulator (70,70,70) of the surgical instrument.

For EndoVis-T, the ground truth is provided as a text file where clasper angle, center point, and shaft axis of the instrument for every frame is listed (see figure 5.2).

Robotic Test Data

The robotic test data consist of 720×576 px frames of additional video material for each of the four surgeries provided for training and two additional recorded surgeries with 1500 frames per video.

The ground truth for testing is provided the same way as for training.

The challenge guidelines specify two methods for testing the trained models. The first method is a *leave-one-surgery-out fashion (LOSO)*: For each of the four surgeries provided for training, one model is trained with the remaining three surgeries. The performance

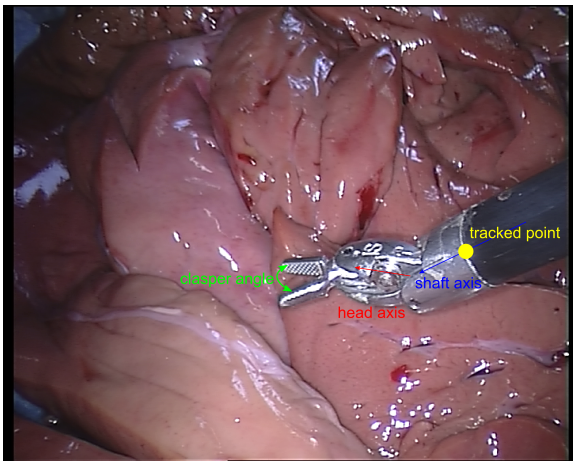
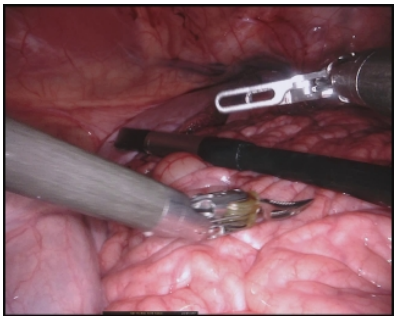


Figure 5.2: Example image of EndoVis15-T provided by the challenge administrators to demonstrate the ground truth annotation for localization of the surgical instruments.



(a) Preprocessed example image.

(b) Preprocessed example mask: The three different instruments have the same colour.

Figure 5.3: Example data provided for EndoVis17-S.

of the model is tested with the additional testing material for the one surgery that was not used for training. This is similar to k-fold cross validation (see also section 2.1.3) with $k = 4$.

The other testing method consists of training one model with the four surgeries provided for training and testing the performance with the two additional surgeries provided for testing.

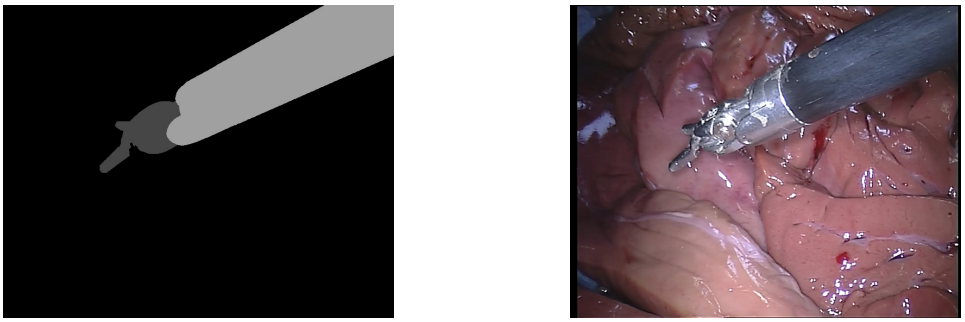
5.2 Data Preprocessing

To prepare the datasets for training and testing, several modifications have to be made to fit them for the network.

5.2.1 Endoscopic Vision Challenge 2017

The images of EndoVis17-S are surrounded by a black canvas (see figure 5.1b). It is removed by cropping the images. This step is proposed in Shvets et al. [15] Without canvas, the images have a resolution of 1280×1024 px. The cropped images are downsampled to a resolution of 320×265 px to decrease training time.

Like the corresponding input images, the ground truth masks have a black canvas that has to be removed. For the part recognition task of EndoVis-S (see figure 5.1a), the



(a) Original example mask with part annotations. (b) Original example image.

Figure 5.4: Unmodified example data provided for EndoVis15-S.

proposed masks have different greyscale values. As the networks in this work solve a binary segmentation problem, the greyscale masks are modified: The different greyscale values are reduced to two values (R,G,B): Background (0,0,0) and instrument (255,255,255). For the type distinction task, every instrument has one single mask. When there is more than one instrument visible at the same time, their masks are merged into one image. After that, one binary mask contains the segmentation ground truth of all instruments shown in the corresponding input image. Like the corresponding input images, the masks are downsampled to a resolution of 320×265 px (see figure 5.3b).

5.2.2 Endoscopic Vision Challenge 2015

The first preprocessing step for EndoVis15-S is extracting the images and masks out of the proposed videos. The resulting input images (see figure 5.4a) and masks (see figure 5.4b) have a resolution of 720×576 px. In order to give the EndoVis15-S data to the same network as the EndoVis17-S data, EndoVis15-S input images and corresponding masks are also downsampled to the resolution of 320×265 px.

To prepare the masks for the binary segmentation task, the same adaptions as for EndoVis17-S are necessary: The masks with greyscale values are converted into a binary image: Every pixel containing an instrument has the (R,G,B) value (255,255,255). The background pixels have the (R,G,B) value (0,0,0). When two instruments are shown at the same time in one input image, the corresponding separated masks for each instrument are merged into one mask.

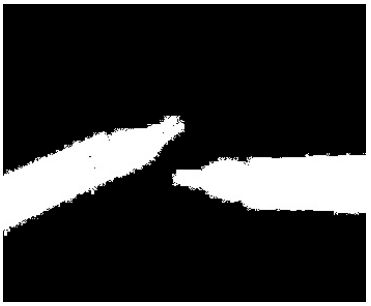
After the conversion to binary masks, erroneous white artefacts surrounding the instrument occur (see figure 5.5a). When applying the operation opening (see Haralick et al. [53]), the white artefacts are reduced to such a low level that the masks can be used for training (see figure 5.5b).

The localization ground truth of the surgical videos is provided as a text file with instrument positions. These instrument positions are each converted into a heatmap, see section 4.3: The standard deviation σ (see equation 4.5) is set to 25. This way, the resulting Gaussian distribution around the center point of the instrument does not exceed the instrument itself, but is larger than a small point (see figure 5.6).

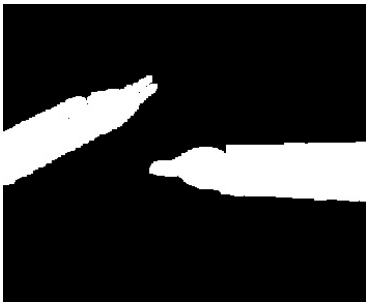
Because the text file with the instrument center points is given for the original images with a size of 720×576 px, the resulting heatmaps have the same size. To fit them to the downsampled input images, the resolution of the heatmaps is reduced to 320×265 pixels.

5.3 Experiments

After preprocessing the proposed datasets, TernausNet-11 (see section 4.1) and LocNet (see section 4.2) are trained and evaluated in different ways with EndoVis15-S, EndoVis17-S,

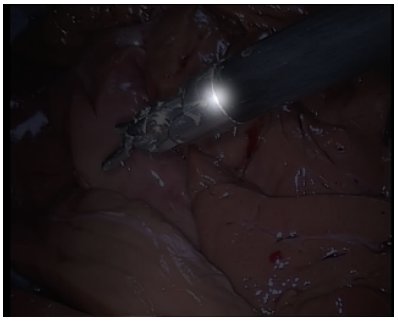


(a) After converting the ground truth images of EndoVis15-S to binary masks, they are surrounded by white artefacts.



(b) After applying opening operation, the amount of white artefacts surrounding the instruments is reduced.

Figure 5.5: Binary masks out of EndoVis15-S before and after the application of opening operation [53].



(a) Example image with superimposed corresponding heatmap.

(b) Example image containing two instruments with superimposed corresponding heatmap.

Figure 5.6: Example images out of EndoVis15-T with corresponding heatmaps.

Segmentation Results TernausNet-11	TER-15S	TER-15S-17S
epochs	200	300
precision	92.77 ± 6.35	92.72 ± 7.51
b.acc.	89.27 ± 3.54	88.26 ± 7.07
accuracy	97.14 ± 1.35	96.94 ± 2.04
specificity	99.32 ± 0.62	99.28 ± 0.68
recall	79.22 ± 6.78	77.23 ± 14
IoU	74.93 ± 8.41	73.39 ± 14.29
dice	85.37 ± 6.15	83.73 ± 11.32

Table 5.1: Results of the different models for the segmentation of surgical instruments. Epochs denote the number of epochs for training in total. For evaluation metrics see section 4.5. The evaluation results are given as mean value over all prediction results $\pm$ the standard deviation.

and EndoVis15-T.

5.3.1 Instrument Segmentation

Because TernausNet-11 is the basis network of LocNet, the segmentation task is evaluated on TernausNet-11 before using the evaluation results to improve the localization task. The training and testing of the models trained with EndoVis15-S is performed according to the challenge guideline LOSO (see section 5.1.2) of EndoVis15. All models were trained with batch-size 4.

The model *TER-15S* is trained for 100 epochs with learning rate $\lambda = 0.0001$ and 100 epochs more with $\lambda = 0.00001$ on EndoVis15-S.

The model *TER-15S-17S* is trained for 100 epochs with $\lambda = 0.0001$ and 100 epochs with $\lambda = 0.00001$ on EndoVis15-S. Afterwards it is trained for 100 epochs with $\lambda = 0.00001$ on EndoVis17-S. For the training on EndoVis17-S, the eight proposed training sets are divided in four subsets. Each of these subsets contains two training sets. These four subsets are used for training and testing the TER-15S-17S model with LOSO fashion on EndoVis15-S.

For the results of both TernausNet-11 segmentation models see table 5.1.

The achieved results are comparable to other state-of-the-art approaches. TER-15S-17S was trained additionally with data from EndoVis17-S in order to test the impact of additional training data on the prediction performance of LocNet. The results for TER-15S and TER-15S-17S for testing on EndoVis15-S are in the same range, i.e. the additional data does not have an impact with high significance on the prediction capability of LocNet for EndoVis15-S. As the EndoVis17-S data are obtained during in-vivo procedures with seven different instruments, they differ from the EndoVis15-S data that were obtained in ex-vivo procedures with less different instrument types. Therefore, the information learned by the model out of EndoVis17-S does not have an improving effect on the prediction for EndoVis15-S.

5.3.2 Concurrent Instrument Segmentation and Localization

Each model is trained and evaluated according to one of the EndoVis15-T challenge guidelines: LOSO fashion (see section 5.1.2), or training on all train sets and then testing on the test sets. For the specific training description of each model see table 5.2.

Training Conditions LocNet			
	epochs	val.method	γ
LOC-01	600	LOSO	1/2
LOC-02	600	LOSO	2/3
LOC-03	50	LOSO	1/2
LOC-04	500	T4	1/2
LOC-05	100	T4	1/2
LOC-06	400	LOSO	1
LOC-07	100	T4	1
LOC-08	500	T4	1

Table 5.2: Validation method (val.method) specifies if the model was trained according to the LOSO fashion (LOSO), or trained completely on the four training sets and tested on the two test sets (T4). The impact of the segmentation and the localization loss on the model is adjusted by γ (see section 4.2).

Segmentation Results LocNet		
	b.acc.	Dice
LOC-01	88.45 ± 1.8	85.25 ± 2.4
LOC-02	$\mathbf{90.65 \pm 1.7}$	$\mathbf{87.8 \pm 2.3}$
LOC-03	89.75 ± 4.1	86.83 ± 6.2
LOC-04	87.45 ± 2	83.92 ± 2.8
LOC-05	89.45 ± 2	86.32 ± 2.77

Table 5.3: Results of the different models for segmentation of surgical instruments. For evaluation metrics see section 4.5. The evaluation results are given as mean value over all prediction results ± the standard deviation. As LOC-06, LOC-07, and LOC-08 are trained with loss factor $\gamma = 1$, the segmentation results for these models are optimized for localization only.

Localization Results Datasets LOSO					
	Dataset1 left/right instr.	Dataset2	Dataset3	Dataset4	mean dist.
LOC-01	$21.24 \pm 12.5/17 \pm 10.4$	13.45 ± 5.9	12.73 ± 5.7	14.22 ± 23.5	16.27 ± 19.6
LOC-02	$22.65 \pm 31.1/15.5 \pm 22.5$	-	-	-	-
LOC-03	$\mathbf{17.58 \pm 12.1/15.15 \pm 9.4}$	$\mathbf{11.36 \pm 6.4}$	$\mathbf{10.83 \pm 6.3}$	$\mathbf{12.05 \pm 8.9}$	$\mathbf{13.85 \pm 10.0}$
LOC-06	$372.24 \pm 91.7/299.62 \pm 251.1$	27.66 ± 109.9	20.47 ± 82.13	21.48 ± 80.2	152.5 ± 207.9

Table 5.4: Results for the different datasets for each LOSO LocNet model. All models were trained according to the LOSO fashion (LOSO). Mean dist. is the mean distance over all datasets. Dataset1 is distinguished into left and right instrument (left/right instr.). The evaluation results are given as mean value over all prediction results ± the standard deviation. The distance between ground truth location and predicted location of the instrument is given in pixels.

	Localization Results Datasets T4		
	Dataset5 left/right instr.	Dataset6 left/right instr.	mean dist.
LOC-04	89.80 ± 101.1/$\mathbf{117.91 \pm 69.0}$	$\mathbf{88.17 \pm 91.2}$/$\mathbf{120.02 \pm 68.2}$	$\mathbf{106.4 \pm 80.4}$
LOC-05	$\mathbf{69.59 \pm 51.6}$/118.15 ± 73.2	100.15 ± 118.8/120.32 ± 70.0	111.7 ± 94.5
LOC-07	114.13 ± 136.8/121.51 ± 78.8	215.36 ± 225/122.23 ± 73.5	162.04 ± 163.9
LOC-08	121.5 ± 146.5/122.43 ± 80.9	231.56 ± 229.3/123.5 ± 77.5	169.7 ± 169.7

Table 5.5: Results for the different datasets for each T4 LocNet model. T4 denotes that the models were trained completely on the four training sets and tested on the two test sets. Each dataset is distinguished into left and right instrument (left/right instr.). Mean dist. is the mean distance over all datasets. The evaluation results are given as mean value over all prediction results ± the standard deviation. The distance between ground truth location and predicted location of the instrument is given in pixels.

For the segmentation results of all LocNet models see table 5.3, for the localization results see table 5.4 and table 5.5.

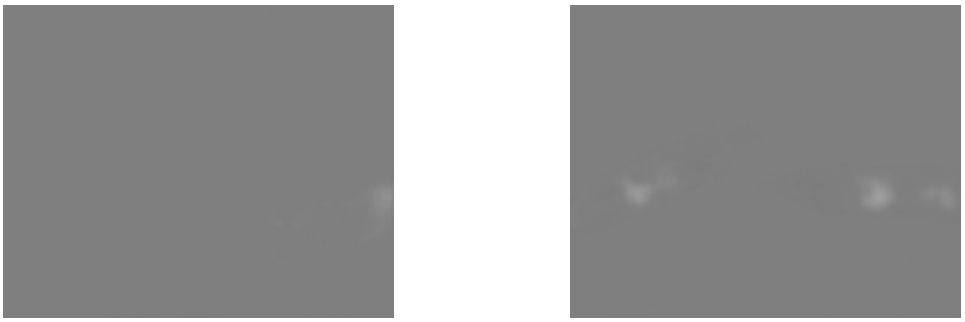
As the instrument positions are extracted from the heatmaps with k-means clustering and k is set to the number of instruments expected to be seen in the input frame, when the number of instruments actually seen in the input frame differs, the localization results are distorted. Another problem is that when the center point of an instrument is disappearing for a certain period of frames, but not the complete instrument, the prediction of the model will still contain higher pixel values, yielding a false-positive localization result for the partly visible instrument. This is because the model predicts a Gaussian distribution surrounding the center point, so even when the center point is not visible, the spread of the Gaussian is contained in the prediction (see figure 5.7a). In this case, comparing the estimated instrument location to the ground truth leads to unreasonably bad results, since the ground truth location for a center point not shown in the image is $(-1, -1)$ and not the actual instrument location. For this reason, the frames where the center point of the instruments is missing according to the ground truth annotation are excluded from evaluation.

To solve the problems in general, further postprocessing steps could be taken: By checking the distance between the cluster center points, the case when two expected heatmaps are collapsed into one could be recognized, which would indicate that only one of the two instruments is visible. Setting a threshold for the amount of higher pixel values found in the predicted heatmap to recognize frames where the center point is probably missing, could solve the general disappearing instrument problem.

5.4 Discussion

The models trained according to LOSO fashion vary in number of epochs and the loss factor γ. LOC-01 and LOC-03 only differ in the number of epochs. The evaluation results are in the same range, but LOC-03 trained with 50 epochs outperforms LOC-01 which is trained with 600 epochs. The resulting heatmaps for LOC-01 look different from LOC-03 for Dataset1, the spread of the Gaussian distribution contains a higher variance for LOC-01 (see figure 5.7b).

This indicates that LOC-03 does not find the center point with high certainty, but the resulting location values are still slightly better than for LOC-01. This could be because the distribution of the Gaussians around the center points are still separated sufficiently, so it is possible to determine the center points with k-means clustering with a high accuracy.



(a) Example heatmap prediction where the instrument center point is not visible in the input image.

(b) Example heatmap prediction with high variance.

Figure 5.7: Example of two heatmap predictions: (a) Heatmap prediction where the instrument center point is disappearing. Notably, the prediction does only contain a low range of pixel values because of the missing center point. (b) Heatmap prediction of LOC-01 with high variance.

As LOC-03 is trained with less epochs than LOC-01, this could indicate that LOC-03 generalizes better and LOC-01 is overfitted on the training data. LOC-02 differs from LOC-01 only in the size of γ: The impact of the localization loss is 2/3, the impact of the segmentation loss is 1/3. The resulting values for Dataset1 are slightly worse than for LOC-01 and LOC-03. The resulting heatmaps contain Gaussian distributions with high variance. For LOC-06, γ is set to 1, therefore the impact of the segmentation loss is set to 0. It is trained 350 epochs more than LOC-02. For the LOSO method, this model achieves the lowest results, especially for Dataset1. As Dataset1 is the only dataset where two instruments are shown, and therefore LOC-06 tested on this dataset has not seen more than one instrument during training, it is possible that the missing segmentation information makes it more difficult for the network to predict a previously unseen number of instruments.

The models trained according to T4 fashion vary in the number of epochs and the loss factor γ. LOC-04 differs from LOC-05 in the number of epochs, it is trained 400 epochs more than LOC-05. γ is set to four for both models, therefore the segmentation and localization loss have the same impact. The results for both models are in the same rage. The models LOC-07 and LOC-08 are trained with $\gamma = 1$, the impact of the segmentation is set to 0. LOC-08 is trained with 400 epochs more than LOC-07. The results for LOC-08 are lower than for the other proposed T4 models. The evaluation results for the T4 models differ from the results for the LOSO models with an inaccuracy of more than 70 pixels. Notably, the ground truth annotations for the test datasets Dataset5 and Dataset6 seem to be inaccurate: This is stated by the challenge administrators [12], an example can be seen in figure 5.8.

In summary, the best LOSO model is LOC-03 and the best T4 model is LOC-04. This indicates that setting γ to 0.5 yields models with good prediction results.

5.4.1 Comparison to State of the Art Methods

The values for segmentation evaluation are in the same range as the current state-of-the-art results, except for the models trained with loss factor $\gamma = 1$.

For training method LOSO, the evaluation results for localization outperform the state-of-the-art methods with the models LOC-01 and LOC-03. For training method T4, the evaluation results are lower or similar to the state-of-the-art methods.

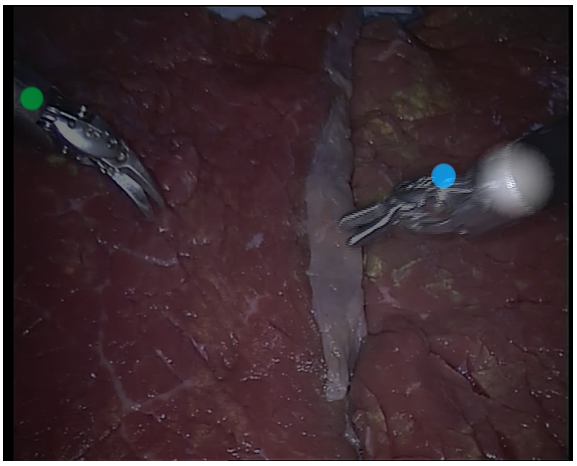


Figure 5.8: Example input image with superimposed predicted preprocessed heatmap by LOC-07. The given ground truth for the left instrument is denoted in green, for the right instrument in blue. Notably, the center point of the left instrument is not visible, and the center point of the right instrument is actually positioned right from the ground truth annotation.

5.4.2 Future Work

Besides taking further postprocessing steps to improve results, the proposed method could be combined with a temporal tracking algorithm. In this case, the estimated instrument locations could serve as input to trackers such as Kalman filter or Particle filter [54]. Temporal tracking would be especially helpful to deal with occlusions and to associate each heatmap with either the left or the right instrument, even when instruments cross.

6. Conclusion

In this work, the goal was to obtain a method for segmentation and localization of surgical instruments in endoscopic video frames. To solve this task, the segmentation CNN TernausNet-11 [15] is extended in order to predict localization of surgical instruments in endoscopic videos. This approach was recently proposed by Laina et al. [12]. All models were trained and evaluated according to LOSO and T4 fashion of the EndoVis15 challenge (see section 5.1.2),

For the segmentation task, all models trained with loss factor $\gamma < 1$ achieved similar results. The results are in range of the current state-of-the-art methods. LOC-02 achieved the best segmentation results.

All models trained according to the challenge guideline LOSO achieved better results than the models trained according to the T4 guidelines. The reason is probably that the ground truth annotations for the proposed test datasets are inaccurate, therefore correct predictions of the models are evaluated worse as they actually are. The model LOC-03 achieved the best prediction results for the LOSO training procedure, it outperforms the current state-of-the-art methods for this evaluation method. The model LOC-04 achieved the best prediction results for the T4 training procedure. The evaluation results are lower than for current state-of-the-art methods for this evaluation method.

The proposed method could be used in combination with a temporal tracking algorithm in order to serve as input to trackers like Kalman filter or Particle filter. Temporal tracking would be helpful to deal with occlusions and associate each heatmap in the endoscopic video to the left or right instrument, even when the instrument are crossing.

List of Figures

List of Tables

Bibliography

[1] A. Darzi and S. Mackay, "Recent advances in minimal access surgery," *BMJ*, pp. 31–34, 2002.

[2] S. Blausen, "Medical gallery of Blausen Medical 2014," *WikiJournal of Medicine*, 2014.

[3] J. Leven, D. Burschka, R. Kumar, G. Zhang, S. Blumenkranz, X. D. Dai, M. Awad, G. D. Hager, M. Marohn, M. Choti, C. Hasser, and R. H. Taylor, "DaVinci Canvas: A Telerobotic Surgical System with Integrated, Robot-Assisted, Laparoscopic Ultrasound Capability," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005*, J. S. Duncan and G. Gerig, Eds., 2005, pp. 811–818.

[4] D. Bouget, M. Allan, D. Stoyanov, and P. Jannin, "Vision-based and marker-less surgical tool detection and tracking: a review of the literature," *Medical Image Analysis*, pp. 633 – 654, 2017.

[5] J. C. Rutherford, M. Stowasser, T. J. Tunny, S. A. Klemm, and R. D. Gordon, "Laparoscopic adrenalectomy," *World Journal of Surgery*, pp. 758–761, 1996.

[6] A. M. Lacy, J. C. García-Valdecasas, S. Delgado, A. Castells, P. Taurá, J. M. Piqué, and J. Visa, "Laparoscopy-assisted colectomy versus open colectomy for treatment of non-metastatic colon cancer: a randomised trial," *The Lancet*, pp. 2224 – 2229, 2002.

[7] O. A. J. van der Meijden and M. P. Schijven, "The value of haptic feedback in conventional and robot-assisted minimal invasive surgery and virtual reality training: a current review," *Surgical Endoscopy*, pp. 1180–1190, 2009.

[8] M. Allan, S. Ourselin, S. Thompson, D. J. Hawkes, J. Kelly, and D. Stoyanov, "Toward Detection and Localization of Instruments in Minimally Invasive Surgery," *IEEE Transactions on Biomedical Engineering*, pp. 1050–1058, 2013.

[9] V. Lahanas, C. Loukas, and E. Georgiou, "A simple sensor calibration technique for estimating the 3d pose of endoscopic instruments," *Surgical Endoscopy*, 2016.

[10] R. Elfring, M. de la Fuente, and K. Radermacher, "Assessment of optical localizer accuracy for computer aided surgery systems," *Computer Aided Surgery*, pp. 1–12, 2010.

[11] Y. Hu, H. U. Ahmed, C. Allen, D. Pendsé, M. Sahu, M. Emberton, D. Hawkes, and D. Barratt, "MR to Ultrasound Image Registration for Guiding Prostate Biopsy and Interventions," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009*, G.-Z. Yang, D. Hawkes, D. Rueckert, A. Noble, and C. Taylor, Eds., 2009, pp. 787–794.

[12] 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," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 664–672, 2017.

[13] S. Bodenstedt, M. Allan, A. Agustinos, X. Du, L. C. García-Peraza-Herrera, H. Kenngott, T. Kurmann, B. P. Müller-Stich, S. Ourselin, D. Pakhomov, R. Sznitman, M. Teichmann, M. Thoma, T. Vercauteren, S. Voros, M. Wagner, P. Wochner, L. Maier-Hein, D. Stoyanov, and S. Speidel, "Comparative evaluation of instrument segmentation and tracking methods in minimally invasive surgery," *CoRR*, 2018.

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[15] A. Shvets, A. Rakhlin, A. A. Kalinin, and V. Iglovikov, "Automatic Instrument Segmentation in Robot-Assisted Surgery Using Deep Learning," 2018.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105.

[17] W. S. Sarle, "Neural Networks and Statistical Models," 1994.

[18] J. Ding, B. Chen, H. Liu, and M. Huang, "Convolutional Neural Network With Data Augmentation for SAR Target Recognition," *IEEE Geoscience and Remote Sensing Letters*, pp. 364–368, 2016.

[19] S. ichi Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, pp. 185 – 196, 1993.

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, 2015.

[21] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, 1986.

[22] J. M. Hernández-Lobato and R. P. Adams, "Probabilistic backpropagation for scalable learning of bayesian neural networks," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

[23] M. Buscema, "Back propagation neural networks," *Substance use & misuse*, pp. 233–270, 1998.

[24] L. Prechelt, "Automatic early stopping using cross validation: quantifying the criteria," *Neural Networks*, pp. 761 – 767, 1998.

[25] A. Y. Ng *et al.*, "Preventing" overfitting" of cross-validation data," in *ICML*, 1997, pp. 245–253.

[26] D. Marmanis, M. Datcu, T. Esch, and U. Stilla, "Deep learning earth observation classification using imagenet pretrained networks," *IEEE Geoscience and Remote Sensing Letters*, pp. 105–109, 2016.

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, pp. 2278–2324, 1998.

[28] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems*, 1990, pp. 396–404.

[29] X. Du, T. Kurmann, P.-L. Chang, M. Allan, S. Ourselin, R. Sznitman, J. D. Kelly, and D. Stoyanov, "Articulated multi-instrument 2D pose estimation using fully convolutional networks," *IEEE transactions on medical imaging*, 2018.

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *MICCAI*, pp. 234–241, 2015.

[31] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *CoRR*, 2014.

[32] L. C. García-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 *Computer-Assisted and Robotic Endoscopy*, T. Peters, G.-Z. Yang, N. Navab, K. Mori, X. Luo, T. Reichl, and J. McLeod, Eds., 2017, pp. 84–95.

[33] D. Pakhomov, V. Premachandran, M. Allan, M. Azizian, and N. Navab, "Deep Residual Learning for Instrument Segmentation in Robotic Surgery," *CoRR*.

[34] M. Attia, M. Hossny, S. Nahavandi, and H. Asadi, "Surgical tool segmentation using a hybrid deep CNN-RNN auto encoder-decoder," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, pp. 3373–3378.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[36] C. Rupprecht, C. Lea, F. Tombari, N. Navab, and G. D. Hager, "Sensor substitution for video-based action recognition," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 5230–5237.

[37] J. Schwartz, A. Steger, and A. Weißl, "Fast Algorithms for Weighted Bipartite Matching," in *Experimental and Efficient Algorithms*, 2005, pp. 476–487.

[38] A. Chaurasia and E. Culurciello, "LinkNet: Exploiting encoder representations for efficient semantic segmentation," in *2017 IEEE Visual Communications and Image Processing (VCIP)*, 2017, pp. 1–4.

[39] A. Agustinos and S. Voros, "2D/3D Real-Time Tracking of Surgical Instruments Based on Endoscopic Image Processing," in *Computer-Assisted and Robotic Endoscopy*, X. Luo, T. Reichl, A. Reiter, and G.-L. Mariottini, Eds., 2016, pp. 90–100.

[40] S. Bodenstedt, M. Wagner, B. Mayer, K. Stemmer, H. Kenngott, B. Müller-Stich, R. Dillmann, and S. Speidel, "Image-based laparoscopic bowel measurement," *International Journal of Computer Assisted Radiology and Surgery*, pp. 407–419, 2016.

[41] K. P. Murphy *et al.*, "Naive bayes classifiers," *University of British Columbia*, 2006.

[42] V. Iglovikov, S. Mushinskiy, and V. Osin, "Satellite imagery feature detection using deep convolutional neural network: A kaggle competition," *CoRR*, 2017.

[43] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Computer Vision, Graphics, and Image Processing*, pp. 355 – 368, 1987.

[44] D. A. Sipkins, X. Wei, J. W. Wu, J. M. Runnels, D. Côté, T. K. Means, A. D. Luster, D. T. Scadden, and C. P. Lin, "In vivo imaging of specialized bone marrow endothelial microdomains for tumour engraftment," *Nature*, p. 969, 2005.

[45] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," 2011.

[46] K. O. Babalola, B. Patenaude, P. Aljabar, J. Schnabel, D. Kennedy, W. Crum, S. Smith, T. F. Cootes, M. Jenkinson, and D. Rueckert, "Comparison and evaluation

of segmentation techniques for subcortical structures in brain mri," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2008*, D. Metaxas, L. Axel, G. Fichtinger, and G. Székely, Eds., 2008, pp. 409–416.

[47] C. E. Metz, "Basic principles of ROC analysis," *Seminars in Nuclear Medicine*, pp. 283 – 298, 1978.

[48] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The Balanced Accuracy and Its Posterior Distribution," in *2010 20th International Conference on Pattern Recognition*, 2010, pp. 3121–3124.

[49] "MICCAI Endoscopic Vision Challenge: Subchallenge Robotic Instrument Segmentation," https://endovissub2017-roboticinstrumentsegmentation.grand-challenge.org, accessed: 2018-09-15.

[50] L. Morelli, S. Guadagni, G. Di Franco, M. Palmeri, G. Caprili, C. D'Isidoro, R. Pisano, A. Moglia, V. Ferrari, G. Di Candio *et al.*, "Use of the new Da Vinci Xi® during robotic rectal resection for cancer: technical considerations and early experience," *International journal of colorectal disease*, pp. 1281–1283, 2015.

[51] "MICCAI Endoscopic Vision Challenge: Subchallenge Instrument Segmentation and Tracking," https://endovissub-instrument.grand-challenge.org/, accessed: 2018-09-15.

[52] G. Watanabe and N. Ishikawa, "Da vinci surgical system," *Kyobu geka. The Japanese journal of thoracic surgery*, pp. 686–689, 2014.

[53] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image Analysis Using Mathematical Morphology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 532–550, 1987.

[54] R. G. Brown, P. Y. Hwang *et al.*, *Introduction to random signals and applied Kalman filtering*. Wiley New York, 1992.