Unsupervised temporal context learning using convolutional neural networks for laparoscopic workflow analysis

Sebastian Bodenstedt[a,*], Martin Wagner[b], Darko Katic[a], Patrick Mietkowski[b], Benjamin Mayer[b], Hannes Kenngott[b], Beat Müller-Stich[b], Rüdiger Dillmann[a], Stefanie Speidel[a]

[a]*Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe*
[b]*Department of General, Visceral and Transplant Surgery, University of Heidelberg, Heidelberg*

Abstract

Computer-assisted surgery (CAS) aims to provide the surgeon with the right type of assistance at the right moment. Such assistance systems are especially relevant in laparoscopic surgery, where CAS can alleviate some of the drawbacks that surgeons incur. For many assistance functions, e.g. displaying the location of a tumor at the appropriate time or suggesting what instruments to prepare next, analyzing the surgical workflow is a prerequisite. Since laparoscopic interventions are performed via endoscope, the video signal is an obvious sensor modality to rely on for workflow analysis.

Image-based workflow analysis tasks in laparoscopy, such as phase recognition, skill assessment, video indexing or automatic annotation, require a temporal distinction between video frames. Generally computer vision based methods that generalize from previously seen data are used. For training such methods, large amounts of annotated data are necessary. Annotating surgical data requires expert knowledge, therefore collecting a sufficient amount of data is difficult, time-consuming and not always feasible.

In this paper, we address this problem by presenting an unsupervised method for training a convolutional neural network (CNN) to differentiate between laparoscopic video frames on a temporal basis. We extract video frames at regular intervals from 324 unlabeled laparoscopic interventions, resulting in a dataset of approximately 2.2 million images. From this dataset, we extract image pairs from the same video and train a CNN to determine their temporal order. To solve this problem, the CNN has to extract features that are relevant for comprehending laparoscopic workflow.

Furthermore, we demonstrate that such a CNN can be adapted for surgical workflow segmentation. We performed image-based workflow segmentation on a publicly available dataset of 7 cholecystectomies and 9 colorectal interventions.

Keywords: Laparoscopy, workflow analysis, convolutional neural network, pretraining, video segmentation, phase detection
2010 MSC: 00-01, 99-00

1. Introduction

The aim of a computer-assisted surgery system (CAS) is to provide the surgeon with the right type of assistance at the right moment. In laparoscopic surgery, such a system could be used to compensate for some of the drawbacks typical to laparoscopy, such as the limited field of view or difficult orientation in the abdominal cavitiy, by e.g. providing assistance during navigation.

For many applications in CAS, such as providing the position of a tumor, specifying the most probable tool required next by the surgeon or determining the remaining duration of surgery, analyzing the surgical workflow is a prerequisite. Since laparoscopic surgeries are performed using an endoscopic camera, a video stream is always available during surgery, making it the obvious choice as input sensor data for workflow analysis. Many workflow analysis tasks, e.g. phase recognition, skill

---

*Corresponding author
Email address: bodenstedt@kit.edu (Sebastian Bodenstedt)

assessment, automatic reporting, video indexing or automatic annotation, require a method for providing a temporal representation of video frames, or rather their content.

Often, laparoscopic tool usage [1][2][3] or surgical activities [4][5][6] are used as feature for such a representation, but currently this information is usually derived through additional hardware (e.g. RFID tags in the case of [2]), which is not generally available in the OR or through manual annotation, which is not feasible for online workflow segmentation or large datasets. The kinematic data from a robotic system, such as the daVinci can be used for providing tool usage information and tool trajectories [7][8], but this information is only available for robotic interventions and not the majority of laparoscopic interventions.

While methods for automatically extracting information on tool usages from endoscopic images do exist [9][10] there are few publications with a purely image-based approach for workflow analysis [9][11][12][13][14]. The authors in [9], [11] and [12] utilize a combination of manually selected image features to describe the content of single video frames. Manually selecting image features has the drawback that only information that the domain expert is aware of can be captured, other characteristics that might still contribute are possibly lost.

In computer vision, one possible solution to the issue of feature selection are convolutional neural networks (CNN), a type of artificial neural network, which has the ability to learn image features. CNNs are currently the state of the art in many areas in computer vision, such as object detection and image classification [15] [16].

In [13], the authors propose EndoNet, a combination of a CNN and a hybrid hidden markov model (HHMM). The CNN here is used to automatically learn image features that can be used to distinguish different surgical phases in laparoscopic gallbladder removals, which are then fed into a HHMM to determine the most probable phase for each image frame. On the dataset of the Endoscopic Vision 2015 Workflow Challenge[1] (EndoVis15Workflow), EndoNet outperforms the method outlined in [11], which uses manually selected image features. The drawback of EndoNet is that a large amount of annotated data is used for training, 40 videos of laparoscopic gallbladder removals in which not only

the surgical phases, but also the laparoscopic instruments are annotated for each frame. This amount of annotated data is difficult and costly to collect. If one takes into consideration that laparoscopic gallbladder removals are simple and standardized operations, one can assume that more complex types of interventions, such as colorectal or pancreatic surgery, would require even more labeled data. In [14], the authors present a CNN-based approach for offline phase detection that outperforms EndoNet on the EndoVis15Workflow dataset, which uses only 6 operations for training. Offline phase detection means that data from the entire intervention is used for assigning a phase to each frame retrospectively. The approaches makes usage of spatio-temporal information to capture object motion during the course of a laparoscopic intervention. The features extracted with the CNN are then combined with either a linear model, a semi-markov model or a time-invariant model, based on dynamic time warping, with the latter two models outperforming [13], leading to the conclusion that including temporal information during workflow analysis improves classification outcome.

One of the advantages of CNNs is that it is possible to take a CNN that is solving one task (e.g. detecting cars) and retrain it for solving a different task (e.g. detecting bicycles) [17]. Retraining (or pretraining), instead of training a new CNN from scratch, has the advantage that previously learned features (say features that respond to wheels) can be repurposed. By repurposing features, a pretrained CNN should require less training data to achieve adequate performance. Generally training deep CNNs requires a large amount of annotated data, which, especially in a surgical environment, is not always feasible to obtain, since usually experts are required to annotate data. Pretraining the CNN using unlabeled data would therefore be preferable.

In [18], the authors train a CNN to develop an understanding of the spatial context of different excerpts from a given image. For this, they divide unlabeled images into multiple 3x3 box grids and train a CNN to arranged the outer blocks correctly in relation to the center block. Part of this trained CNN is then modified and retrained to partake in an object detection challenge, achieving state of the art results. Inspired by [18], we extended the idea of pretraining a CNN with spatial context information to pretraining with temporal context information provided by given videos.

---

[1]http://endovissub-workflow.grand-challenge.org/

In this paper, we propose a method for a CNN to learn visual features by sorting frames from videos of laparoscopic interventions into the correct temporal order. We assume that the features learned while solving the sorting task enable the CNN to distinguish frames based on their temporal context. Such a CNN can be used as starting point for many applications were these visual cues would be beneficial, e.g. online and offline video segmentation, automatic annotation, indexing and generating surgical reports. This temporal context learning task is performed using unlabeled laparoscopic videos.

Furthermore, we evaluate the suitability of such a pretraining for a supervised workflow segmentation task, in this case, segmenting surgical videos into phases (surgical phase detection). For this, the pretrained CNN is extended to take information from the current frame and previous frames to deduce the phase of the current frame. In contrast to other approaches, this method does not rely on manually-selected features or expensive annotation of surgical tools and, furthermore, is able to detect phases online. The evaluation is performed on two datasets, the EndoVis15Workflow dataset, which contains 7 annotated cholecystectomies and a dataset containing 9 laparoscopic colorectal interventions, a more complex type of surgery, recorded in the University Hospital of Heidelberg.

2. Unsupervised temporal context learning

In this section, we present our method for training such a deep CNN using unlabeled videos. We accomplish this by solving a task that requires the CNN to sort two given frames into the correct temporal order. For this, a large dataset from multiple laparoscopic interventions is used. We assume that solving such a task requires the CNN to learn to extract visual cues that describe the temporal flow of laparoscopic interventions.

2.1. Training task

The task we propose for training the CNN is illustrated in figure 1: Given two frames from the same laparoscopy, what is the most probable relative order of the two frames, i.e. which frame comes first? We uniformly sample two random frames from the video of a laparoscopic intervention and feed it into our CNN. The CNN must then compute the relative order of the two frames in the original video, i.e. which frame comes first. We assume that solving

Figure 1: Our task for pretraining a CNN. Which is the most probable temporal order of the two images? (Answer: the right image comes first, the clip has to be inserted into the body, before being placed.)

this task requires the CNN to extract visual cues relevant to surgical workflow and thereby develop an understanding of the temporal flow of laparoscopic interventions.

2.2. Dataset

To train the CNN, we used a large dataset consisting of 324 laparoscopic interventions recorded anonymously at the University Hospital of Heidelberg. The dataset contains videos of 30 different types of laparoscopic interventions, providing a diverse range in training data. The videos were all recorded in the same operating room using the integrated operating room system OR1^{TM} (Karl Storz GmbH & Co KG, Tuttlingen, Germany). The interventions were performed by multiple surgeons with varying endoscopes and optics. We extracted frames at intervals of one frame per second, resulting in approximately 2.2 million images. Since the videos were recorded automatically, we had to ensure that sequences that did not contain any large changes (e.g. black screens) were excluded from the dataset. This was accomplished by excluding a video frame f from the dataset, if for the last video frame g from the same video that was included in the dataset

$$||I(f) - I(g)|| < 8000$$

, with $I(f)$ and $I(g)$ being the respective pixel values for each image.

2.3. Training the CNN

Selecting a network topology that allows a CNN to predict the relative order of two given video frames from scratch can be a difficult task. We therefore based our model on the one presented in [18], which was shown to work for spatial context prediction. The topology of the network used can be seen in fig. 2. A pair of frames from the same
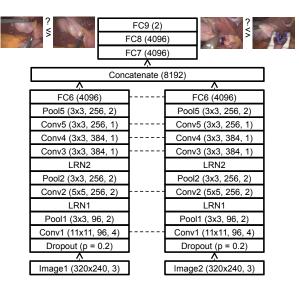
Figure 2: CNN Topology for the temporal context prediction task: Dotted lines indicate shared weights. *Dropout* are dropout layers that, with a probability of p, set a value to zero. *Conv* are convolutional layers, *LRN* are local response normalization layers[16], *Pool* are max-pooling layers, *FC* are fully connected layers and *Concatenate* concatenate two input vectors. The numbers in parenthesis indicate size of filter kernel, number of outputs and step size. In the case of fully connected layers, the number of hidden units is listed instead.

video is fed into the two input layers of the CNN. Each frame is then processed by a chain of multiple convolutional layers (Conv1 to Conv5), each with AlexNet-style topology [16], resulting in a reduced representation of the frame in a fully connected layer (FC6). The corresponding layers in both chains share weights. The outputs of the two FC6 layers are then concatenated and then processed using two further fully connected layer. FC9 then outputs if either frame 1 (Output: 0) oder frame 2 (Output: 1) comes first in a temporal order. For every convolutional and fully connected layer, except FC8, a ReLu (rectified linear unit) nonlinearity[19] was used. FC9 uses a softmax nonlinearity instead.

During training, for each epoch (iteration) we sample with replacement 256 operations out of all operations. From each of these operations, 3 frames, I_1, I_2 and I_3, are drawn randomly, with $I_t < I_{t+1}$ or, in other words, I_t precedes I_{t+1} in a temporal order. The frames are then resampled to a resolution of 320×240. To ascertain that the proportions inside the frames are not skewed by this, we crop the borders of the images to give the image a $4:3$ aspect ratio in case they exhibited a different

ratio. Furthermore, we normalize each value in the RGB channels by mapping them into the range of $[-0.5, 0.5]$. We then form 6 inequations, i.e.

$$I_0 < I_1, \ I_0 < I_2, \ I_1 < I_2$$

$$I_1 > I_0, \ I_2 > I_0, \ I_2 > I_1$$

resulting in 1536 inequations per epoch. The CNN is then trained for 10000 epochs using stochastic gradient descent (learning rate of 0.0005) combined with nesterov momentum (momentum of 0.9). As loss function, we selected categorical cross-entropy. The CNN was implemented in Python, using Theano[20] and Lasagne[21], and trained using NVidia GTX Titan X and NVidia GTX 1080.

3. Laparoscopic workflow segmentation

For a given laparoscopic frame, the method outlined in section 2 provides a descriptor that makes a temporal distinction possible. In this section, we determine the suitability of such a descriptor for surgical workflow segmentation, i.e. dividing a given surgical in coherent and semantic meaningful segments.

3.1. NaiveLWFNet

A naive approach to workflow segmentation would be to extend one of the processing chains (everything before FC6) with further fully-connected layers to assign each frame to the most probable class label. We constructed a naive CNN for laparoscopic workflow analysis (*NaiveLWFNet*, Naive **L**aparoscopic **W**ork**F**low **Net**work) as can be seen in fig. 3.

While distinguishing frames certainly is a prerequisites for laparoscopic phase detection, determining the current state from just a single frame seems questionable and prone to ambiguities. We assume that single frames alone do not contain sufficient information to deduce the current phase and therefore propose to extent *NaiveLWFNet* to include information seen in previous frames.

3.2. LWFNet

Feedforward neural networks, by definition, do not contain cycles and therefore do not recollect previous states to compute the current output. Recurrent neural networks (RNN) overcome this limitation by introducing cycles in the topology of the network and thereby allowing the network to
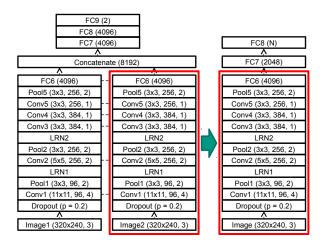
Figure 3: *NaiveLWFNet*: For a naive approach to laparoscopic phase detection, we take part of the CNN illustrated in fig. 2 and add two further fully-connected layers to assign a class. Here N indicates the number of phases.

Figure 4: *LWFNet*: To incorporate previously seen information into our approach for workflow segmentation, we *NaiveLWFNet* and combine it with a gated recurrent unit (GRU)[23], which makes it possible to retain information from previous frames.

process sequences. Tradition RNNs suffer from multiple drawbacks, such as gradients that vanish over the course of training and recalling only "recent" information [22]. Long term-short term memory units (LSTM)[22], a deep RNN architecture, do not suffer from these drawbacks and, furthermore, are selective about the information they retain and forget. Similar to LSTMs, gated recurrent units (GRU)[23] also do not suffer from the drawbacks of traditional RNN architecture and can learn to recall/forget particular information. Seeing as GRUs perform similarly to LSTMs for certain tasks[24], whilst having fewer parameters, we decided to extend *NaiveLWFNet* with a GRU (fig. 4) into *LWFNet*. To integrate the GRU into *LWFNet*, the output from FC6 has to be modified slightly, as RNNs expect sequences as input. For this, we reshape the output from FC6, a 2D tensor of the shape $batchsize \times 4096$, to a 3D tensor of shape $1 \times batchsize \times 4096$, simulating a $batchsize$ long sequence. Generally, the number of frames in a video exceeds the batch size, meaning that, instead of one long sequence, the GRU only sees multiple shorter sequences. To compensate for this, we take the contents of the hidden state after the last element of the sequence and use it to initialize the hidden state before processing the next batch.

3.3. Training

The CNN is trained using stochastic gradient descent (initial learning rate λ_0 was set to 10^{-3}) combined with nesterov momentum (momentum of 0.9)
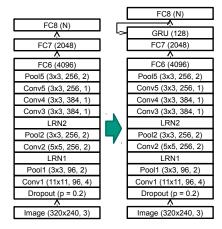
for multiple epochs with a batch size of 256. To penalize large weights and thereby prevent overfitting, we apply L1 and L2 regularization during training. For this, we add terms to the cost function, which incorporate the L1 and L2 norm of the weights and thereby penalize large weights, lowering the risk of overfitting. We selected a weight of 10^{-5} for the L1 penalty term and 10^{-3} for the L2 penalty term. To ensure convergence, we reduced the learning rate λ by factor α: $\lambda_{t+1} = \alpha \cdot \lambda_t$. For α, we selected 0.975 as value. Since we are only interested in fine-tuning the parameters learned in section 2, we use a smaller learn rate $\lambda_t' = 10^{-1} \cdot \lambda_t$ for FC6 and all layers proceeding it. The value for the parameters specified here were determined empirically.

4. Evaluation

We evaluated the presented approaches for workflow segmentation on two datasets for laparoscopic phase detection. To compare our proposed method to the state of the art, we first evaluate on the publicly available EndoVis15Workflow dataset. Furthermore, to show that our method translates to longer, more complex interventions, we evaluate our method on a dataset comprised of colorectal interventions from the University Hospital of Heidelberg.

4.1. Metrics

The following metrics were used to evaluate the performance of the different workflow segmentation

methods on a given video from a laparoscopic intervention:

- Precision: Percentage of frames correctly attributed to a certain phase

- Recall: Percentage of frames attributed to a certain phase that are correctly attributed to that phase

- Accuracy: Overall percentage of frames attributed to the correct phase

For each analyzed video, we will compute the average over all phases for precision and recall.

4.2. EndoVis15Workflow

The public dataset from the EndoVis 2015 workflow challenge consists of 7 laparoscopic cholecystectomies provided by the Technische Universität München. The videos have been segmented into surgical phases, seven phases in total (tab. 1). For each video frame the corresponding label was provided as annotation. To train both *NaiveLWFNet*

Phase ID	Explanation
1	Placement of trocars
2	Preparation of Calots triangle
3	Clipping and cutting of cystic artery and duct
4	Gallbladder dissection
5	Gallbladder retrieval
6	Hemostasis
7	Attaching drainage, wound closure and end of operation

Table 1: Different phases in EndoVis15Workflow.

and *LWFNet*, we first sampled the provided videos at a rate of one frame per second, in order to reduce the data and thereby the time required for training. We also resampled the resolution of the selected frames from 1920×1080 to 320×240. Using this slightly modified data, we then performed a leave-one-surgery-out evaluation (training on 6 videos and testing on the 7th video for all seven possible combination of training videos). For each test set, we trained for 100 epochs. The development of the accuracies for each run can be found in figure 5. To demonstrate the advantage of the proposed pretraining, we also included results for a version of *LWFNet* with randomly initialized weights in figure 5(c). Figure 5 clearly shows that the GRU

(a) *NaiveLWFNet*

(b) *LWFNet*

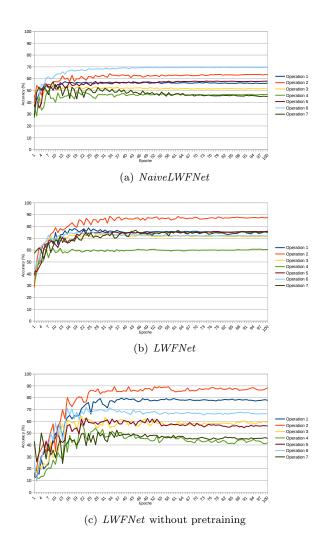(c) *LWFNet* without pretraining

Figure 5: Development of the accuracies of the phase detection for each operation and for each network during the leave-one-surgery-out evaluation on the EndoVis dataset.

based methods outperform the feedforward based *NaiveLWFNet*. Furthermore, we are also able to demonstrate that pretraining *LWFNet* as outlined in section 2 increases performance when compared to randomly initialized parameters. Table 2 further highlights this, as it shows that *LWFNet* with pretraining achieves a higher precision, recall and accuracy in comparison to *LWFNet* without pretraining and *NaiveLWFNet*.

We also compared our results to those published by Twinanda et al. [13] and Dergachyova et al. [11] (tab. 2). *LWFNet* outperforms the method presented by Dergachyova et al. and the CNN only version of EndoNet. The CNN + HHMM based EndoNet outperforms *LWFNet*, which can be at-

6

	Precision	Recall	Accuracy
NaiveLWFNet	56.6% ± 7.5%	53.7% ± 8.8%	56.3% ± 8.1%
LWFNet	79.3% ± 8.1%	73.7% ± 9.7%	74.5% ± 8.4 %
LWFNet without pretraining	75.4% ± 11.8%	68.8% ± 12.6%	66.0% ± 14.8%
EndoNet (CNN only)[13]	64.8% ± 7.3%	64.3% ± 11.8%	65.9% ± 4.7%
EndoNet (CNN + HHMM)[13]	83.0% ± 12.5%	79.2% ± 17.5%	76.3% ± 5.1%
Dergachyova et al.[11]	72.1% ± 16.4%	71.3% ± 13.6%	68.1%

Table 2: Comparison of the results of our proposed methods, EndoNet [13] (only online results) and the method proposed by Dergachyova et al.[11].

tributed the large task specific dataset used for training EndoNet.

	Precision	Recall	Accuracy
P1	85.2%±12.0%	98.3%±4.4%	98.6%±1.1%
P2	81.8%±8.3%	89.0%±9.6%	94.0%±5.2%
P3	72.0%±25.8%	64.0%±34.0%	89.3%±5.1%
P4	71.7%±34.2%	55.8%±41.6%	88.3%±4.4%
P5	77.5%±23.1%	83.3%±14.1%	92.5%±5.6%
P6	78.4%±23.6%	51.6%±37.9%	88.4%±5.8%
P7	88.4%±15.2%	73.9%±26.7%	97.8%±1.6%

Table 3: Performance of *LWFNet* broken down into the different phases.

Table 3 shows how *LWFNet* performs for each of the 7 phases individually. The phases closes to the start and the finish achieve the highest performance in all metrics, while phase further away perform somewhat worse. Of all phases, phase 6 has the lowest accuracy and recall, which can be attributed to the fact that phase 5 and 6 are often intermingled and visually very similar, making them difficult to distinguish. Phase 4 also has a low performance, which could be explained by mix-ups with phases 3 and 6, which are also visually similar.

4.3. Colorectal laparoscopy

The colorectal dataset consists of 9 colorectal laparoscopies recorded at the University Hospital of Heidelberg. These 9 interventions are made up out of 6 proctocolectomies and 3 rectal resections. While these interventions were recorded in the same manner as the dataset outlined in section 2.2, the two datasets are disjunct. Each of these videos was segmented into 8 phases (see table 4) by the same surgical expert.

Similarly as to the previous section, we extracted one frame per second from the laparoscopic videos and resampled the frames to a resolution of 320 × 240. With this dataset, we then performed a leave-one-surgery-out evaluation for both *NaiveLWFNet* and *LWFNet*. For each test set, we trained for

Phase ID	Explanation
1	Team Time-Out
2	Preparation and orientation at abdomen
3	Mobilization of colon
4	Dissection of lymph nodes and blood vessels
5	Dissection and resection of rectum
6	Preparation of anastomosis
7	Placing stoma
8	Finishing the operation

Table 4: Different phases in the colorectal dataset.

100 epochs. The same evaluation was also performed for a version of *LWFNet* with no pretrained weights. The progression of the accuracies of each test run for each method can be found in figure 6. The graphs clearly show that even for this dataset, the GRU based methods achieve a higher accuracy than *NaiveLWFNet*. As seen in the previous section, the pretraining also boosts the classification performance on this dataset.

	Precision	Recall	Accuracy
NaiveLWFNet	32.0% ± 9.6%	29.7% ± 8.5%	50.4% ± 9.0%
LWFNet	68.2% ± 15.0%	52.6% ± 9.8%	67.2% ± 13.1 %
LWFNet without pretraining	53.9% ± 6.7%	43.6% ± 11.2%	62.8% ± 14.1%

Table 5: Comparison of the results of our proposed methods on the colorectal dataset from the University of Heidelberg.

This assumption is confirmed by table 5. A pretrained *LWFNet* achieves higher values for precision, recall and accuracy than *LWFNet* without pretraining and *NaiveLWFNet*.

	Precision	Recall	Accuracy
P1	88.1%±28.0%	85.8%±30.2%	99.5%±0.7%
P2	72.9%±24.1%	67.0%±33.3%	97.8%±1.4%
P3	72.7%±15.7%	74.8%±31.2%	83.4%±5.5%
P4	58.7%±43.9%	9.3%±17.1%	91.4%±5.8%
P5	76.7%±14.1%	80.3%±18.7%	80.8%±9.7%
P6	57.7%±31.0%	37.0%±37.0%	88.2%±10.2%
P7	55.7%±52.5%	11.5%±33.2%	97.4%±2.3%
P8	62.9%±45.1%	51.3%±42.3%	96.8%±3.5%

Table 6: Performance of *LWFNet* on the colorectal dataset broken down into the different phases.

The phase-wise performance of *LWFNet* is listed in table 6. Phases 4 and 7 achieve the lowest performance. Phase 4 is often so confused with phase 3, which precedes it and phase 5, which generally follows it. Phase 7 is a rather short phase, meaning

(a) *NaiveLWFNet*

(b) *LWFNet*

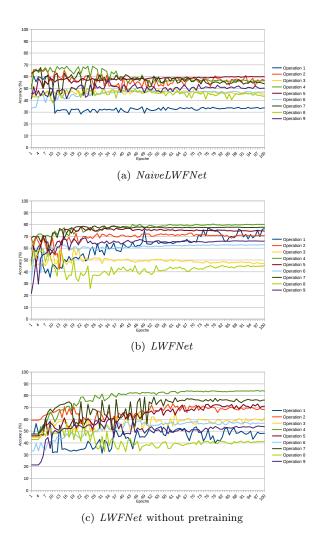
(c) *LWFNet* without pretraining

Figure 6: Development of the accuracies of the phase detection for each operation and for each network during the leave-one-surgery-out evaluation on the colorectal dataset.

only a small number of examples were available for training and visually similar to phase 5 with which it is often confused.

5. Discussion

In this paper, we presented a method that allows us to train a CNN to differentiate between frames taken from the same video in a temporal context. To train such a CNN, only the temporal order between two frames is required, which can be inferred from a given video. Therefore no additional manual annotations by a surgical expert are required

Furthermore, we showed that such a network can be adapted to solve certain video segmen-

tation tasks, in particular surgical phase detection. We evaluated the method on two datasets: a publicly available dataset of annotated cholecystectomies and a dataset of annotated colorectal interventions. The evaluation showed that on both datasets a GRU-based approach outperforms a plain feed-forward network. A combination of the GRU-based approach and the pretrained model further increased performance, supporting our hypothesis that applying the pretraining method outlined in section 2 would be beneficial.

Our proposed method, which combines pretraining and a GRU, performs comparable to the state of the art on the public dataset, while the feedforward and the non-pretrained method perform significantly lower. *LWFNet* outperforms the method of Dergachyova et al.[11] and the purely CNN-based EndoNet[13], which did not include temporal information. A second version of EndoNet incorporates temporal information using a hierarchical hidden markov model and thereby achieves a higher performance than *LWFNet*. When comparing the performance of EndoNet and *LWFNet*, one has to take into consideration that EndoNet used 40 further annotated cholecystectomies for training.

A laparoscopic cholecystectomy is a very standardized and simple intervention. Therefore, to show that our method can also be applied to longer, more complex laparoscopic interventions, we performed another evaluation on a dataset consisting of colorectal interventions, which are generally more complex in terms of involved anatomy, vessel resection and required level of surgical expertise. The resulting performance was lower than on the cholecystectomy dataset. This, in our opinion, can be attributed to the large variance in the dataset, which should be expected with long and complex interventions. The order of certain phases varied partially between different interventions, e.g. in operation 7 phase 7 was not performed and in most operations, phase 3 was interrupted multiple times by other phases. This can be partially attributed to the fact that the interventions were performed by different surgeons, as different surgeons have different preferences when it comes to the order of certain parts of the procedure. The endoscopic optic and the tools used also varied between interventions. This leads us to conclude that more examples, which mirror this variance, are required to increase performance. Nevertheless, we were able to show that our pretrained CNN achieves a higher performance on this dataset than a randomly initialized CNN. To im-

8

prove the result of the phase segmentation, further post-processing steps, e.g. a sliding window or a hidden markov model, could be applied to the output of our CNN for smoothing.

In addition to surgical phase detection, the pre-trained network could possible be used for other tasks in laparoscopy. One application could be other segmentation tasks, such as action detection or event recognition. Furthermore, the of output of layer fc6 could be used as a reduced representation of a laparoscopic frame for allowing indexing of surgical videos.

References

[1] L. Bouarfa, P. Jonker, J. Dankelman, Discovery of high-level tasks in the operating room, Journal of Biomedical Informatics 44 (3) (2011) 455 – 462, biomedical Complexity and Error. doi:http://dx.doi.org/10.1016/j.jbi.2010.01.004. URL http://www.sciencedirect.com/science/article/pii/S1532046410000055

[2] R. Stauder, A. Okur, L. Peter, A. Schneider, M. Kranzfelder, H. Feussner, N. Navab, Random Forests for Phase Detection in Surgical Workflow Analysis, Springer International Publishing, Cham, 2014, pp. 148–157. doi:10.1007/978-3-319-07521-1_16. URL http://dx.doi.org/10.1007/978-3-319-07521-1_16

[3] N. Padoy, T. Blum, S.-A. Ahmadi, H. Feussner, M.-O. Berger, N. Navab, Statistical modeling and recognition of surgical workflow, Medical Image Analysis 16 (3) (2012) 632 – 641, computer Assisted Interventions. doi:http://dx.doi.org/10.1016/j.media.2010.10.001. URL http://www.sciencedirect.com/science/article/pii/S1361841510001131

[4] D. Katić, A.-L. Wekerle, F. Gärtner, H. Kenngott, B. P. Müller-Stich, R. Dillmann, S. Speidel, Knowledge-Driven Formalization of Laparoscopic Surgeries for Rule-Based Intraoperative Context-Aware Assistance, Springer International Publishing, Cham, 2014, pp. 158–167. doi:10.1007/978-3-319-07521-1_17.

URL http://dx.doi.org/10.1007/978-3-319-07521-1_17

[5] T. Neumuth, G. Strauß, J. Meixensberger, H. U. Lemke, O. Burgert, Acquisition of Process Descriptions from Surgical Interventions, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 602–611. doi:10.1007/11827405_59. URL http://dx.doi.org/10.1007/11827405_59

[6] G. Forestier, L. Riffaud, P. Jannin, Automatic phase prediction from low-level surgical activities, International Journal of Computer Assisted Radiology and Surgery 10 (6) (2015) 833–841. URL http://dx.doi.org/10.1007/s11548-015-1195-0

[7] R. DiPietro, C. Lea, A. Malpani, N. Ahmidi, S. S. Vedula, G. I. Lee, M. R. Lee, G. D. Hager, Recognizing surgical activities with recurrent neural networks, CoRR abs/1606.06329. URL http://arxiv.org/abs/1606.06329

[8] L. Zappella, B. Béjar, G. Hager, R. Vidal, Surgical gesture classification from video and kinematic data, Medical Image Analysis 17 (7) (2013) 732 – 745, special Issue on the 2012 Conference on Medical Image Computing and Computer Assisted Intervention. doi:http://dx.doi.org/10.1016/j.media.2013.04.007. URL http://www.sciencedirect.com/science/article/pii/S1361841513000522

[9] T. Blum, H. Feußner, N. Navab, Modeling and Segmentation of Surgical Workflow from Laparoscopic Video, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 400–407. doi:10.1007/978-3-642-15711-0_50. URL http://dx.doi.org/10.1007/978-3-642-15711-0_50

[10] S. Speidel, J. Benzko, S. Krappe, G. Sudra, P. Azad, B. P. Müller-Stich, C. Gutt, R. Dillmann, Automatic classification of minimally invasive instruments based on endoscopic image sequences, Proc. SPIE 7261 (2009) 72610A–72610A–8. doi:10.1117/12.811112. URL http://dx.doi.org/10.1117/12.811112

[11] O. Dergachyova, D. Bouget, A. Huaulmé, X. Morandi, P. Jannin, Automatic data-driven real-time segmentation and recognition of surgical workflow, International Journal of Computer Assisted Radiology and Surgery 11 (6) (2016) 1081–1089. doi:10.1007/s11548-016-1371-x. URL http://dx.doi.org/10.1007/s11548-016-1371-x

[12] F. Lalys, L. Riffaud, D. Bouget, P. Jannin, A framework for the recognition of high-level surgical tasks from video images for cataract surgeries, IEEE Transactions on Biomedical Engineering 59 (4) (2012) 966–976. doi:10.1109/TBME.2011.2181168.

[13] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. de Mathelin, N. Padoy, Endonet: A deep architecture for recognition tasks on laparoscopic videos, CoRR abs/1602.03012. URL http://arxiv.org/abs/1602.03012

[14] C. Lea, J. H. Choi, A. Reiter, G. Hager, Surgical phase recognition: From instrumented ors to hospitals around the world, M2CAI 2016.

[15] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[16] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25, Curran Associates, Inc., 2012, pp. 1097–1105.

[17] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, S. Bengio, Why does unsupervised pretraining help deep learning?, Journal of Machine Learning Research 11 (Feb) (2010) 625–660.

[18] C. Doersch, A. Gupta, A. A. Efros, Unsupervised visual representation learning by context prediction, in: International Conference on Computer Vision (ICCV), 2015.

[19] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: J. Frnkranz, T. Joachims (Eds.), Proceedings of the 27th International Conference on Machine Learning (ICML-10), Omnipress, 2010, pp. 807–814.
URL http://www.icml2010.org/papers/432.pdf

[20] Theano Development Team, Theano: A Python framework for fast computation of mathematical expressions, arXiv e-prints abs/1605.02688.
URL http://arxiv.org/abs/1605.02688

[21] S. Dieleman, J. Schlter, C. Raffel, E. Olson, S. K. Snderby, D. Nouri, D. Maturana, M. Thoma, E. Battenberg, J. Kelly, J. D. Fauw, M. Heilman, diogo149, B. McFee, H. Weideman, takacsg84, peterderivaz, Jon, instagibbs, D. K. Rasul, CongLiu, Britefury, J. Degrave, Lasagne: First release. (Aug. 2015). doi:10.5281/zenodo.27878.
URL http://dx.doi.org/10.5281/zenodo.27878

[22] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[23] K. Cho, B. van Merrienboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, CoRR abs/1409.1259.
URL http://arxiv.org/abs/1409.1259

[24] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555.