



Temporal Coherence-based Self-supervised Learning for Laparoscopic Workflow Analysis

Isabel Funke¹(✉), Alexander Jenke¹, Sören Torge Mees², Jürgen Weitz²,
Stefanie Speidel¹, and Sebastian Bodenstedt¹

¹ Department for Translational Surgical Oncology, National Center for Tumor
Diseases (NCT), Partner Site Dresden, Dresden, Germany
{Isabel.Funke,Alexander.Jenke,Stefanie.Speidel,
Sebastian.Bodenstedt}@nct-dresden.de

² Department of Visceral, Thoracic and Vascular Surgery, Faculty of Medicine,
University Hospital Carl Gustav Carus, TU Dresden, Dresden, Germany

Abstract. In order to provide the right type of assistance at the right time, computer-assisted surgery systems need context awareness. To achieve this, methods for surgical workflow analysis are crucial. Currently, convolutional neural networks provide the best performance for video-based workflow analysis tasks. For training such networks, large amounts of annotated data are necessary. However, collecting a sufficient amount of data is often costly, time-consuming, and not always feasible. In this paper, we address this problem by presenting and comparing different approaches for self-supervised pretraining of neural networks on unlabeled laparoscopic videos using temporal coherence. We evaluate our pretrained networks on Cholec80, a publicly available dataset for surgical phase segmentation, on which a maximum F_1 score of 84.6 was reached. Furthermore, we were able to achieve an increase of the F_1 score of up to 10 points when compared to a non-pretrained neural network.

Keywords: Self-supervised learning · Temporal coherence
Surgical workflow analysis · Surgical phase recognition · Pretraining
CNN-LSTM

1 Introduction

The aim of a computer-assisted surgery (CAS) system is to provide the surgeon with the right type of assistance at the right time. To achieve this, context awareness is crucial. This means that the system must be able to understand the processes currently taking place in the operating room (OR) and adapt its behavior accordingly. *Surgical workflow analysis* covers the challenging topic of perceiving, understanding, and describing surgical processes [11].

I. Funke and S. Bodenstedt—Both authors contributed equally to this work.

A common approach is to analyze surgical processes by interpreting a time series of signals that are recorded by sensors – in some cases also by humans – in the OR. As laparoscopic surgeries are performed via camera, methods that require only video as input sensor data are of special interest, since the video can be collected effortlessly during surgery.

State-of-the-art video-based approaches for workflow analysis rely on deep neural networks [1, 2, 9, 15, 18]. However, deep learning-based methods require large amounts of labeled data for training. Especially in surgery, obtaining a sufficient amount of annotated video data is difficult and costly.

To alleviate the problem of limited training data, it is common to *pretrain* neural networks and fine-tune them afterwards. Often, networks are pretrained using labeled data coming from another domain, such as ImageNet [4]. Another way is to use unlabeled data from the same domain and train on a proxy task using labels inherent in the data, which is called *self-supervised* learning.

For self-supervised learning from video, a number of ideas have been proposed [5, 8, 12, 13, 16]. Most exploit the *temporal coherence* of video, which implies that (i) consecutive frames are in temporal order, (ii) frames change slowly over time, and (iii) frames change steadily, i.e., abrupt motions are unlikely.

The studies [12, 13] propose proxy tasks based on the temporal order between frames. In line with this, [2] use the task to order pairs of laparoscopic images for pretraining a network for surgical phase segmentation. *Surgical phase segmentation* [14] is the problem of recognizing the surgical phase being performed by the surgeon at each point during surgery. Another proxy task for this problem is to predict the progress and remaining duration of a surgery [18].

Intuitively, these tasks encourage the network to learn discriminative features that are useful to infer the absolute or relative temporal position of a video frame. In contrast, [5, 8, 16] aim at learning features that are *invariant* to typical alterations occurring between adjacent frames, such as slight rotations or deformations. To this end, they aim to ensure that temporally close frames, which most likely depict the same semantic scene, are mapped to similar representations in feature space. This idea goes back to *Slow Feature Analysis (SFA)* [17].

In this paper, we describe and compare different approaches to exploit temporal coherence while pretraining a convolutional neural network (CNN) for surgical phase segmentation. We assume the pretraining encourages the CNN to learn features that are invariant to irrelevant changes between adjacent frames, such as slight movements of instruments or of the endoscope, while being discriminative enough to distinguish between semantically different frames.

To promote reproducibility and to fuel future research, we made our code available at https://gitlab.com/nct_tso_public/pretrain_tc.

Experiments using the Cholec80 dataset [15] demonstrate that a CNN pretrained to exploit the temporal coherence of unlabeled laparoscopic video outperforms a non-pretrained CNN after being fine-tuned for surgical phase segmentation. When only 20 labeled videos are available, the proposed pretraining achieves an increase from 67.8 to 78.6 as measured by F_1 score.

2 Methods

The core of our neural network architecture for surgical phase segmentation is a ResNet-50 CNN [6]. We initialize it with ImageNet [4] pretrained weights and further train it on unlabeled videos of laparoscopic surgeries, using an SFA-based approach for self-supervised learning. This encourages the CNN to map temporally close video frames to similar representations in feature space.

More formally, the CNN learns an embedding $f : \mathbb{R}^{3 \times h \times w} \rightarrow \mathbb{R}^d$, where \mathbb{R}^d is the d -dimensional feature space and $\mathbb{R}^{3 \times h \times w}$ is the space of laparoscopic video frames with height h , width w , and three color channels (RGB). Let $I_t \in \mathbb{R}^{3 \times h \times w}$ denote the frame at time step t . To suffice temporal coherence, we require that $f(I_t) \approx f(I_{t+\Delta})$ for a small Δ with $|\Delta| < \delta$. To learn an embedding that is discriminative and to avoid trivial solutions such as $f(I_t) := 0$, we require that $f(I_t)$ and $f(I_{t+\Gamma})$ lie further apart in feature space when Γ is large, i.e., $|\Gamma| > \gamma$ (see Subsect. 2.1 for details). δ and γ are non-negative real-valued parameters.

To evaluate the efficacy of the proposed self-supervised pretraining approach, we extend the CNN into a recurrent neural network (RNN) and fine-tune the CNN-RNN for surgical phase segmentation using annotated laparoscopic videos (see Subsect. 2.2). We can then compare the performance of the pretrained CNN-RNN to the performance of a CNN-RNN that has been trained solely for the surgical phase segmentation task (see Sect. 3).

2.1 Self-supervised Pretraining

For self-supervised pretraining, the output layer of the ResNet-50 CNN is replaced with a fully connected layer with $d = 4096$ output neurons (*FeatureNet*). As the CNN has been pretrained on ImageNet, we only adjust the weights of the *conv5_x* layers and of the newly added fully connected layer during training.

Given a frame I_t , we calculate the embedding $F_t := f(I_t)$ by forwarding the frame through *FeatureNet* and taking the output $(o_1, o_2, \dots, o_d)^T \in \mathbb{R}^d$ at the last layer. We train *FeatureNet* to learn a temporally coherent video frame embedding using one of the following methods. Throughout this section, D denotes a distance function, in our case the L2 norm.

(a) Training with **contrastive** loss

Given a video with T frames, we create a tuple $(I_t, I_{t+\Delta}, I_{t+\Gamma})$ by sampling t from $[0, T-1]$, Δ from $[-\delta, \delta]$, and Γ from $[-(T-1), -\gamma] \cup [\gamma, T-1]$ uniformly at random. Regarding *FeatureNet* as a Siamese network [3], we propagate the temporally close pair $(I_t, I_{t+\Delta})$ through the CNN and calculate $D(F_t, F_{t+\Delta})$. Likewise, we propagate the temporally distant pair $(I_t, I_{t+\Gamma})$ and calculate $D(F_t, F_{t+\Gamma})$. Finally, we calculate the contrastive loss [5]

$$L_c(F_t, F_{t+\Delta}, F_{t+\Gamma}) = D(F_t, F_{t+\Delta}) + \max\{0, m_c - D(F_t, F_{t+\Gamma})\}.$$

This loss function encourages F_t to be close to $F_{t+\Delta}$, while F_t and $F_{t+\Gamma}$ are enforced to be separated by margin m_c .

(b) Training with **ranking** loss

A training tuple $(I_t, I_{t+\Delta}, I_{t+\Gamma})$ is created the same way as in method (a). Regarding *FeatureNet* as a Triplet Siamese Network, we propagate the triplet $(I_t, I_{t+\Delta}, I_{t+\Gamma})$ through the CNN and calculate the ranking loss [16]

$$L_r(F_t, F_{t+\Delta}, F_{t+\Gamma}) = \max\{0, D(F_t, F_{t+\Delta}) - D(F_t, F_{t+\Gamma}) + m_r\}.$$

This loss function considers the distance between F_t and $F_{t+\Delta}$ relative to the distance between F_t and $F_{t+\Gamma}$ and encourages F_t and $F_{t+\Delta}$ to be closer together than F_t and $F_{t+\Gamma}$ by a margin of m_r .

(c) Training with **1st & 2nd order contrastive** loss

While (first order) temporal coherence requires the first order temporal derivatives in the learned feature space to be small, i.e., $F_t \approx F_{t+\Delta}$, *second order temporal coherence* [8] requires the second order temporal derivatives to be small, i.e., $F_t - F_{t+\Delta} \approx F_{t+\Delta} - F_{t+2\Delta}$ for a small value of Δ .

Intuitively, first order temporal coherence ensures that embeddings do not change quickly over time, while second order temporal coherence ensures that the changes are consistent, or steady, across neighboring frames. Applying the contrastive loss function to second order temporal coherence yields

$$L_{c_2}(F_t, F_{t+\Delta}, F_{t+2\Delta}, F_{t+\Gamma}) = L_c(F_t - F_{t+\Delta}, F_{t+\Delta} - F_{t+2\Delta}, F_{t+\Delta} - F_{t+\Gamma})$$

In practice, we create a training tuple $(I_t, I_{t+\Delta}, I_{t+2\Delta}, I_{t+\Gamma})$ by sampling t , Δ , and Γ as described in method (a). Regarding *FeatureNet* as a Triplet Siamese Network, we propagate the triplets $(I_t, I_{t+\Delta}, I_{t+2\Delta})$ and $(I_t, I_{t+\Delta}, I_{t+\Gamma})$ through the network and calculate L_{c_2} . We then combine it with the first order contrastive loss L_c into an overall loss $L_{c+c_2} = L_c + \omega L_{c_2}$, where $\omega = 0.5$ is a non-negative real-valued weight parameter.

2.2 Supervised Fine-Tuning for Surgical Phase Segmentation

Once pretrained, we modify the CNN for surgical phase segmentation by extending it into an RNN using a long short-term memory unit (LSTM) [7] with 512 neurons. The LSTM is followed by a fully connected layer, which has one output neuron per surgical phase. We refer to this CNN-LSTM as *PhaseNet*. During fine-tuning, the weights of the CNN and the LSTM are jointly optimized. However, the weights of the ResNet-50 layers below *conv5_x* stay frozen.

3 Evaluation

For evaluation, we used the publicly available Cholec80 dataset [15]. It consists of 80 videos from laparoscopic cholecystectomies, annotated with surgical phase labels. We divided the dataset into four sets A, B, C, and D of equal size and similar average procedure length. A, B, and C were used for training, while D was withheld for testing. For pretraining, we extracted video frames at 5 Hz.

Training and testing for phase segmentation was performed at 1 Hz. Each frame was downsized to 384×216 px.

We trained three different versions of *FeatureNet*, one with each of the pre-training variants described in Sect. 2.1. The union of sets A, B, and C (i.e., 60 videos in total) was used as training data, ignoring the labels. Each CNN was trained for 25 epochs. Per epoch, we randomly sampled 250 tuples per video, which were processed in batches of size 64. δ was set to 30 s (15 s for variant (c)), γ to 120 s and $m_c = m_r = 2$. We used the Adam optimizer [10] with a learning rate of 10^{-4} . All newly added layers were initialized with random values from the range $(\frac{-1}{\sqrt{n}}, \frac{1}{\sqrt{n}})$, with n being the number of neurons in the layer.

To evaluate the suitability of the proposed pretraining approach for surgical phase segmentation, each of the pretrained CNNs (**contrastive**, **ranking**, and **1st & 2nd order contrastive**) was extended into a *PhaseNet* and fine-tuned using the labeled videos from either set A (#OPs = 20), sets A and B (#OPs = 40), or sets A, B, and C (#OPs = 60). As baseline, a *PhaseNet* without self-supervised pretraining (**no pretraining**) was fine-tuned in the same manner. Note that the underlying ResNet-50 CNN had still been pretrained on ImageNet.

For fine-tuning the networks, we used the Adam optimizer [10] with a learning rate of 10^{-4} and a batch size of 128. After every batch, the content of the LSTM’s hidden state was saved and restored for the next batch. Due to hardware restraints, gradients were only accumulated for three batches before applying the optimizer. Training was stopped once the accuracy on the training set climbed above 99.9%. All newly added layers were initialized as described above.

The results of evaluating each *PhaseNet* on test set D can be found in Table 1. We calculated the metrics *accuracy*, *recall*, and *precision* as defined in [14]. The F_1 score is the harmonic mean of precision and recall. The metrics were

Table 1. Performance of the baseline (first row) and the pretrained models on the surgical phase segmentation task. #OPs denotes how many labeled OPs were used.

	#OPs	Accuracy	Recall	Precision	F_1 score
No pretraining	20	78.8 ± 12.5	72.3 ± 11.4	73.4 ± 12.9	67.8 ± 14.1
	40	88.8 ± 7.7	83.2 ± 8.4	83.8 ± 9.2	80.4 ± 10.3
	60	89.7 ± 6.6	82.8 ± 9.4	85.8 ± 7.6	80.8 ± 10.3
Contrastive	20	84.4 ± 10.6	77.2 ± 8.4	78.8 ± 5.3	73.9 ± 8.9
	40	91.7 ± 5.5	85.4 ± 6.1	88.2 ± 5.6	83.8 ± 7.1
	60	92.0 ± 4.5	86.2 ± 4.2	85.5 ± 4.8	83.6 ± 4.9
Ranking	20	86.1 ± 7.2	79.4 ± 6.5	82.9 ± 5.9	77.2 ± 7.4
	40	90.2 ± 6.4	85.6 ± 6.2	85.2 ± 5.9	82.5 ± 7.3
	60	90.3 ± 5.4	85.2 ± 6.2	86.1 ± 5.2	82.9 ± 6.9
1 st & 2 nd order contrastive	20	88.1 ± 5.8	80.7 ± 5.7	83.8 ± 5.6	78.6 ± 6.1
	40	90.7 ± 10.4	86.3 ± 7.5	86.9 ± 6.1	83.4 ± 10.1
	60	92.7 ± 4.3	87.0 ± 4.0	87.6 ± 5.3	84.6 ± 5.4

averaged over all operations in the test set. Table 2 presents the phase-wise results of the best performing pretrained *PhaseNet* (**1st & 2nd order contrastive**) compared to the *PhaseNet* that did not undergo self-supervised pretraining.

Table 2. Comparison of the baseline and the best performing pretrained model. We report the average F_1 scores calculated for each of the phases P1 to P7.

	#OPs	P1	P2	P3	P4	P5	P6	P7
No pre-training	20	64.5±35.7	83.4±15.9	59.0±33.4	80.8±14.2	62.0±24.0	62.8±20.2	62.4±18.3
	40	88.7±21.6	92.3±10.8	75.3±29.4	90.3±14.3	74.0±17.6	71.7±19.4	71.0±14.3
	60	82.4±25.3	94.9±5.7	81.1±18.5	92.0±10.9	76.1±15.3	73.7±17.9	65.4±24.6
1 st & 2 nd order contrastive	20	79.3±25.7	92.2±7.6	81.9±14.4	91.3±8.0	72.4±17.5	72.8±17.5	60.0±24.4
	40	87.6±15.3	95.7±7.1	86.3±13.3	91.4±18.1	78.5±18.4	73.2±21.7	71.6±18.1
	60	90.2±14.7	97.6±2.7	89.3±9.9	95.9±3.7	75.4±19.3	76.9±18.2	67.8±16.6

4 Discussion

Table 1 clearly shows that all three pretrained models outperform the baseline when being fine-tuned on the same set of labeled training data. The performance boost is especially apparent when only 20 labeled videos are available. Here, in terms of F_1 score, pretraining achieves an increase from 67.8 to up to 78.6 while halving the standard deviation. Pretraining still improves performance when more labeled videos are available. Notably, the pretrained models fine-tuned on only 40 labeled videos outperform the baseline trained on 60 videos. We conclude that the proposed SFA-based pretraining enables a CNN to learn feature representations that are beneficial to the task of surgical phase segmentation.

Comparing the three pretraining variants, we do not find big differences. All in all, using a combination of first and second order temporal coherence for pretraining seems to offer the largest boost to performance, especially when only few (20) labeled videos are used.

Looking at the results with respect to each surgical phase (Table 2), we see that most phases benefit greatly from pretraining (variant **1st & 2nd order contrastive**) when only 20 labeled videos are available. The effect of pretraining diminishes when the number of labeled videos is increased, but is still noticeable in the majority of phases. Only the benefit to phase P7 seems negligible.

P7 contains visual similarities with P5 and P6, which makes them difficult to distinguish. Since the phase is short (about 1 to 3 min), frames that we label as close during pretraining may belong to previous phases. Likewise, frames that belong to previous phases but are temporally close are not selected as distant pair. Hence, the network learns features that are rather invariant than discriminative with regard to phase P7 and P6 or P5.

To shed some light on the features learned during pretraining, we investigated which images the network considers similar. We selected query frames $\{I^q\}$ from a video used during pretraining. Then, for each frame I^q and each video v in the test set, we identified the frame $I^{q,v}$ in v that is most similar to I^q , i.e., closest

to I^q in feature space. More formally, $I^{q,v} = \operatorname{argmin}_{I_t \in v} D(f(I^q), f(I_t))$, where D was chosen to be the L2 norm. To calculate the embedding f , we used the **1st & 2nd order contrastive** pretrained *FeatureNet* (before fine-tuning).

Figure 1 presents four selected queries. Generally, it can be seen that images that are close in feature space show similar scenes with regard to anatomical structures and/or tool presence. The first and second query frames depict scenes that only differ in the amount of blood visible, a trait also observed in the retrieved frames. Likewise, the third and fourth query frames show similar scenes. However, the third query frame is unusual as the specimen bag is closed. Observing that the retrieved images are semantically not closely related to the query frame, we assume that its embedding does not reflect the presence of the specimen bag. For the fourth query frame, which is visually similar but more representative, semantically similar frames are retrieved.

We refrain from comparing temporal coherence-based learning to other pre-training methods for surgical phase segmentation [2, 18] since these studies were conducted using other datasets, namely EndoVis2015 (7 cholecystectomies) in [2] and 120 cholecystectomies in [18].

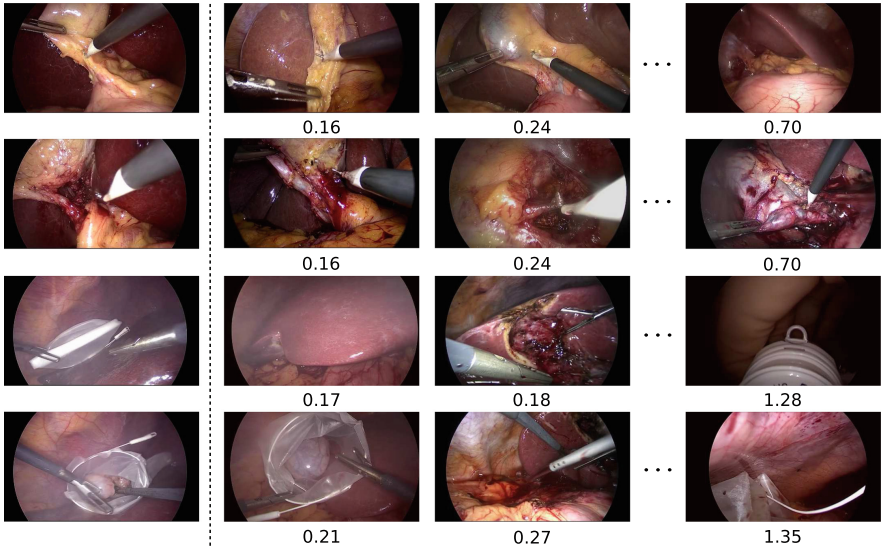


Fig. 1. Image retrieval task. Each row represents one query. Left-most: Query frame. Right: The frames closest in feature space, one per test video. Numbers denote distance to query frame. The depicted frames are sorted with regard to this distance.

5 Summary

In this paper, we show that the temporal coherence of unlabeled laparoscopic video can be exploited for self-supervised pretraining by training a CNN to map

temporally close video frames onto embeddings that are close in feature space. When extended into a CNN-LSTM architecture for surgical phase segmentation, all pretrained models outperform the non-pretrained baseline when being fine-tuned on the same labeled dataset. Using a combination of first and second order temporal coherence, the pretrained models even perform similarly or better than the baseline when less labeled data is used. Combining our approach with temporal order-based concepts into a more holistic temporal coherence-based pretraining method could possibly enhance the discriminative properties of the learned embedding and improve performance even further.

Future work will address the question whether the learned embeddings can be used for unsupervised detection of more fine-grained video segments, such as surgical activities or steps. Furthermore, we will investigate whether the notion of slow and steady features is beneficial for regularization during supervised training compared to using the concept during a separate pretraining phase.

References

1. Aksamentov, I., Twinanda, A.P., Mutter, D., Marescaux, J., Padoy, N.: Deep neural networks predict remaining surgery duration from cholecystectomy videos. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D.L., Duchesne, S. (eds.) MICCAI 2017. LNCS, vol. 10434, pp. 586–593. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66185-8_66
2. Bodenstedt, S., et al.: Unsupervised temporal context learning using convolutional neural networks for laparoscopic workflow analysis. arXiv preprint [arXiv:1702.03684](https://arxiv.org/abs/1702.03684) (2017)
3. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a “siamese” time delay neural network. In: NIPS, pp. 737–744 (1994)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR, pp. 248–255 (2009)
5. Goroshin, R., Bruna, J., Tompson, J., Eigen, D., LeCun, Y.: Unsupervised learning of spatiotemporally coherent metrics. In: ICCV, pp. 4086–4093 (2015)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
8. Jayaraman, D., Grauman, K.: Slow and steady feature analysis: higher order temporal coherence in video. In: CVPR, pp. 3852–3861 (2016)
9. Jin, Y., et al.: SV-RCNet: workflow recognition from surgical videos using recurrent convolutional network. *IEEE Trans. Med. Imaging* **37**(5), 1114–1126 (2018)
10. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (2015)
11. Lalys, F., Jannin, P.: Surgical process modelling: a review. *Int. J. Comput. Assist. Radiol. Surg.* **9**(3), 495–511 (2014)
12. Lee, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Unsupervised representation learning by sorting sequences. In: ICCV, pp. 667–676 (2017)
13. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and learn: unsupervised learning using temporal order verification. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 527–544. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_32

14. Padoy, N., Blum, T., Ahmadi, S.A., Feussner, H., Berger, M.O., Navab, N.: Statistical modeling and recognition of surgical workflow. *Med. Image Anal.* **16**(3), 632–641 (2012)
15. Twinanda, A.P., Shehata, S., Mutter, D., Marescaux, J., de Mathelin, M., Padoy, N.: EndoNet: a deep architecture for recognition tasks on laparoscopic videos. *IEEE Trans. Med. Imaging* **36**(1), 86–97 (2017)
16. Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. In: *ICCV*, pp. 2794–2802 (2015)
17. Wiskott, L., Sejnowski, T.J.: Slow feature analysis: unsupervised learning of invariances. *Neural Comput.* **14**(4), 715–770 (2002)
18. Yengera, G., Mutter, D., Marescaux, J., Padoy, N.: Less is more: surgical phase recognition with less annotations through self-supervised pre-training of CNN-LSTM networks. arXiv preprint [arXiv:1805.08569](https://arxiv.org/abs/1805.08569) (2018)