Learning Representations of Endoscopic Videos to Detect Tool Presence Without Supervision

David Z. Li[1](✉), Masaru Ishii[2], Russell H. Taylor[1], Gregory D. Hager[1], and Ayushi Sinha[3]

[1] Department of Computer Science, The Johns Hopkins University, Baltimore, USA
dli44@alumni.jhu.edu

[2] Johns Hopkins Medical Institutions, Baltimore, USA

[3] Laboratory for Computational Sensing and Robotics, The Johns Hopkins University, Baltimore, USA

Abstract. In this work, we explore whether it is possible to learn representations of endoscopic video frames to perform tasks such as identifying surgical tool presence without supervision. We use a maximum mean discrepancy (MMD) variational autoencoder (VAE) to learn low-dimensional latent representations of endoscopic videos and manipulate these representations to distinguish frames containing tools from those without tools. We use three different methods to manipulate these latent representations in order to predict tool presence in each frame. Our fully unsupervised methods can identify whether endoscopic video frames contain tools with average precision of 71.56, 73.93, and 76.18, respectively, comparable to supervised methods. Our code is available at https://github.com/zdavidli/tool-presence/.

Keywords: Endoscopic video · Tool presence · Representation learning · Variational autoencoder · Maximum mean discrepancy

1 Introduction

Despite the abundance of medical image data, progress in learning from such data has been impeded by the lack of labels and the difficulty in acquiring accurate labels. With increase in minimally invasive procedures [28], an increasing number of endoscopic videos (Fig. 1) are available. This can open up the opportunity for video-based surgical education and skill assessment. Prior work [18] has shown that both experts and non-experts can produce valid objective skill assessment via pairwise comparisons of surgical videos. However, watching individual videos is time consuming and tedious. Therefore, much work is being done in automating skill assessment using supervised [5] and unsupervised [4] learning. These prior methods used kinematic data from tools to learn surgical motion. However, many endoscopic procedures do not capture kinematic data.

T. Syeda-Mahmood et al. (Eds.): ML-CDS 2020/CLIP 2020, LNCS 12445, pp. 54–63, 2020.
https://doi.org/10.1007/978-3-030-60946-7_6

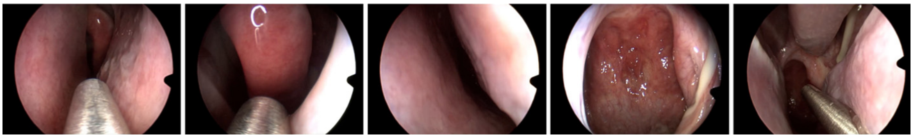


Fig. 1. Examples of endoscopic video frames with and without tools. These examples show the variance in anatomy in our dataset.

Therefore, we want to explore whether we can work towards automated skill assessment directly from endoscopic video.

Since videos contain more information than just kinematics, we want to first isolate tool motion from camera motion in endoscopic videos. If the two types of motion can be disentangled, then representations of video frames with and without tools should be distinct enough to allow separation between the two. Therefore, our aim in this work is to evaluate whether we can detect frames that contain tools. In order to do this, we use a variational autoencoder (VAE) [14] to learn latent representations of endoscopic video frames since VAEs have the ability to learn underlying low-dimensional latent representations from complex, high-dimensional input data. Specifically, we use maximum mean discrepancy (MMD) VAE [30], which uses a modified objective function in order to avoid overfitting and promote learning a more informative latent representation.

We then manipulate these learned representations or encodings using three different methods. First, we directly use the encodings produced by our MMD-VAE to evaluate whether encodings of frames with and without tools can be separated. Second, we model the tool presence as a binary latent factor and train a Bayesian mixture model to learn the clusters over our encodings and classify each frame as containing or not containing tools. Third, we use sequences of our encodings to perform future prediction and evaluate whether temporal context can better inform our prediction of tool presence. Our evaluation methods identify frames containing tools with average precision of 71.56, 73.93, and 76.18, respectively, without any explicit labels.

2 Prior Work

Prior work has shown that surgical motion can be learned from robot kinematics. Lea et al. [16] and DiPietro et al. [5] showed that supervised learning methods can accurately model and recognize surgical activities from robot kinematics. DiPietro et al. [4] further showed that encodings learned from robot kinematics in an unsupervised manner also clustered according to high-level activities. However, these methods rely on robot kinematics which provide information like gripper angle, velocity, etc. Endoscopic procedures that do not use robotic manipulation do not produce kinematics data, but do produce endoscopic videos.

Much work has also been dedicated to extracting tools from video frames using supervised tool segmentation and tool tracking methods. Many methods ignore the temporal aspect of videos and compute segmentation on a frame-by-frame basis. Several methods use established network architectures, like U-Net, to compute segmentations [20,24]. Some methods have tried to tie in the temporal aspect of videos by using recurrent neural networks (RNNs) to segment tools [1], while others have combined simpler fully convolutional networks with optical flow to capture the temporal dimension [7]. More recently, unsupervised methods for learning representations of videos have also been presented [25]. While Srivastava et al. [25] use RNNs to encode sequences of video frames, which can grow large quickly, we will explore whether unsupervised learning of video representations on a per-frame basis will give us sufficient information to discriminate between frames with and without tools.

3 Method

3.1 Dataset

We use a publicly-available sinus endoscopy video consisting of five segments of continuous endoscope movement from the front of the nasal cavity to the back [6]. The video was initially collected at 1080p resolution and split into individual frames. Frames that depicted text or where the endoscope was outside the nose were discarded. A total of 1551 frames, downsampled from 1080p resolution to a height of 64 pixels and centrally cropped to 64×64 pixels, were extracted. This downsampling was necessary due to GPU limitations.

We held out 20% of the frames, sampled from throughout our video sequence, as the test set. Each frame was manually labeled for tool presence. These annotations were used for evaluation only. In the training set, 65.8% of frames were labeled as containing a tool, and in the test set 67.4% of frames were labeled as containing a tool.

3.2 Variational Autoencoder

We use variational autoencoders (VAEs) [14] to learn low dimensional latent representations that can encode our endoscopic video data. VAEs and their extensions [30] are based on the idea that each data point, $x \in \mathbf{X}$, is generated from a d-dimensional latent random variable, $z \in \mathbb{R}^d$, with probability $p_\theta\,(x|z)$, where z is sampled from the prior, $p_\theta(z) \sim \mathcal{N}(\mu,\, \sigma^2)$, parameterized by θ [14]. However, since optimizing over the probability density function (PDF) P is intractable, the optimization is solved over a simpler PDF, Q, to find q_ϕ that best approximates p_θ [14]. To ensure that q best approximates p, θ and ϕ are jointly optimized by maximizing the evidence lower bound (ELBO) [14]:

$$\log p(x) \geq E_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - \mathrm{KL}\left(q_\phi(z|x)\,\|\,p(z)\right). \tag{1}$$

In order to encourage VAEs to learn more informative encodings without overfitting to the data, Zhao et al. [30] introduced the maximum mean discrepancy (MMD) VAE, which maximizes the mutual information between the data and the encodings. MMD-VAE changes the objective function by replacing the KL-divergence term with MMD and introduces a regularization term, λ [30]:

$$\log p(x) \geq E_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - \lambda \text{MMD}\left(q_\phi(z|x) \,\|\, p(z)\right). \tag{2}$$

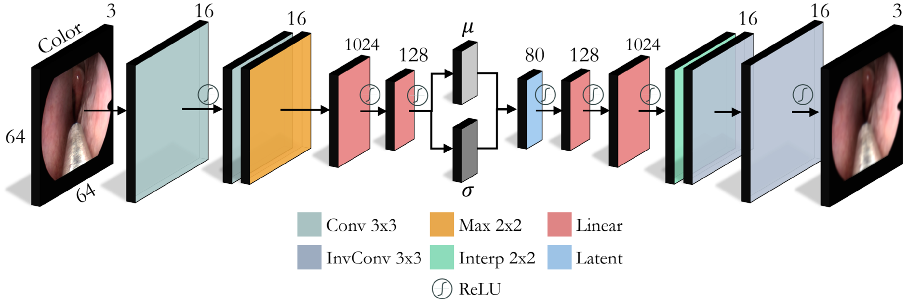


Fig. 2. Our MMD-VAE architecture with a two-layer CNN encoder and decoder.

3.3 Training

We used a convolutional neural network encoder and decoder each with two convolutional layers and three fully-connected layers (Fig. 2). We performed a hyperparameter sweep over latent dimension and regularization coefficient of our MMD-VAE implementation in PyTorch [21] and evaluated each model based on the criteria in Sect. 3.4.

The best performing model from our sweep was trained for 80 epochs using stochastic gradient descent (SGD) on a single NVIDIA Quadro K620 GPU with 2 GB memory with minibatch size of 32 and Adam optimizer [13] with default parameters except for learning rate which was set to 10^{-3}. This model had a latent dimension $d = 20$ and regularization coefficient $\lambda = 5$.

3.4 Model Evaluation

Direct Evaluation. First, we directly evaluate the encodings produced by our MMD-VAE implementation using a query-based evaluation. We compute the cosines between encodings of each test frame, i, containing a tool and all other test frames, $j \neq i$, and threshold the products to separate high and low responses. Since the query (i.e., test frame, i) contains a tool, all other test frames containing tools should produce high response, while those without tools should produce low response. To evaluate our results, we compute the average precision (AP) score [31] over responses from each test frame. AP summarizes the precision-recall curve by computing the weighted mean of precision values computed at each threshold, weighted by the increase in recall from the previous threshold. In simpler terms, AP computes the area under the precision-recall curve.

Approximate Inference. Next, we evaluate our encodings using approximate inference. We estimate $p\,(\text{tool presence}\,|\,\text{latent encoding})$ by modeling the space of encodings as a finite mixture model with a categorical latent variable C which has $K = 2$ Gaussian states (tool present and not present). We use Markov chain Monte Carlo (MCMC) sampling [19] to approximate the posterior distribution by constructing a Markov chain whose states are assignments of the model parameters and whose stationary distribution is p:

$$p(z_i|\boldsymbol{\theta}, \mu_{c_i}, \sigma_{c_i}) = \sum_{c_i=1}^{K} \boldsymbol{\theta}_{c_i}^{\top} \mathcal{N}(z_i|\mu_{c_i}, \sigma_{c_i}). \tag{3}$$

Here, $z_i \in \mathbb{R}^d$ is the ith encoding generated from our MMD-VAE, $d \in \mathbb{N}$ is the dimension of the encoding sample, and each of the K configurations are normally distributed according to parameters $\boldsymbol{\mu}, \boldsymbol{\sigma} \in \mathbb{R}^K$ and mixing probabilities $\boldsymbol{\theta} \in \mathbb{R}^{d \times K}$. We assume each encoding z_i is generated by $\boldsymbol{\theta}$ and a latent state $1 \leq c_i \leq K$, described by $\mathcal{N}(\mu_{c_i}, \sigma_{c_i})$. By running the Markov chain for B burn-in steps, we reach the stationary distribution, p.

We then sample the chain for N iterations which form samples from p [19]. The parameters of the mixture model, $\boldsymbol{\mu}, \boldsymbol{\sigma}$, and $\boldsymbol{\theta}$, are learned using the No-U-turn sampler [10] on Eq. 3. Finally, for a fixed z_i, we can estimate the probability that encoding z_i comes from latent cluster c_i to predict tool presence: $p(c_i|z_i) \propto p(z_i|c_i)p(c_i) = p(z_i, c_i)$.

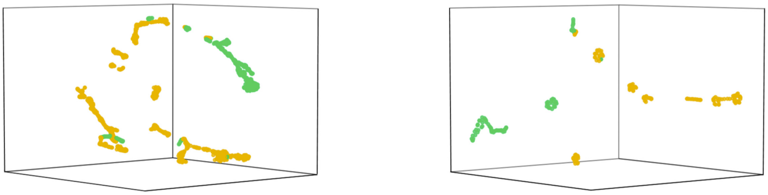
For evaluation, we learned the parameters in PyStan [26] with four Markov chains with $B = N = 2500$ and default hyperparameters. The posterior probability of each sample belonging to each cluster was then computed and used to predict labels. We evaluate the separation between the two latent states and compute the AP score for our label predictions.

Future Prediction. Finally, we evaluate our encodings by training a future prediction model using a recurrent neural network (RNN) encoder-decoder [2] to observe a sequence of past video frame encodings and reconstruct a sequence of future frame encodings [4]. The intuition behind this approach is that models capable of future prediction must encode contextually relevant information [4]. Both the encoder and decoder have long short-term memory (LSTM) [8,9] architectures to avoid the vanishing gradient problem, and each frame of the future sequence is associated with its own mixture of multivariate Gaussians in order not to blur distinct futures together under a unimodal Gaussian [4].

Our PyTorch [21] implementation of the future prediction model was similar to that presented by DiPietro et al. [4]. We used 5 frame sequences of past and future encodings, and Adam [13] for optimization at a learning rate of 0.005 and other hyperparameters at their default values. The latent dimension was set to 64, the number of Gaussian mixture components to 16, and the model was trained for 1000 epochs with a batch size of 50.

As in direct evaluation, we evaluate the encodings produced by future prediction by computing the cosines between encodings from each test sequence,

s_i, containing a tool and all other test sequences, $s_j \neq s_i$, taking the maximum per-frame, and thresholding, as before, to separate high and low responses. A per-frame maximum is computed here since each frame belongs to multiple adjacent sequences, s_j, producing multiple responses. Specifically, since we used 5 frame sequences, each frame produces 5 responses, of which we pick the maximum. Finally, we compute the AP over responses from each test sequence.



(a) MMD-VAE

(b) MMD-VAE + FP

Fig. 3. 3D dimensionality reductions, obtained using t-SNE, of (a) 80D encodings produced by MMD-VAE, and (b) 64D encodings produced by MMD-VAE + FP. MMD-VAE + FP shows slightly better separation between tool (green) and no tool (orange). The labels are used for visualization only. (Color figure online)

4 Results

The results from our three experiments are described in this section. Here, the direct evaluation will be referred to by MMD-VAE, approximate inference method by MMD-VAE + MCMC, and future prediction by MMD-VAE + FP.

We also compare our unsupervised methods to frame-level tool presence predictions from 4 *supervised* methods. Twinanda et al. [29] use a supervised CNN based on AlexNet [15] to perform tool presence detection in a multi-task manner. Sahu et al. [23] use a transfer learning approach to combine ImageNet [3] features with time-series analysis to detect tool presence. Raju et al. [22] combine features from GoogleNet [27] and VGGNet [12] for tool presence detection. Jin et al. [11] use a supervised region-based CNN to spatially localize tools and use these detections to drive frame-level tool presence detection. Results are summarized in Table 1.

Direct Evaluation. The encodings produced by our implementation of MMD–VAE show some amount of separation (Fig. 3a). Therefore, we expect our queries to produce high response when evaluated against frames with tools. However, we also expect queries to produce higher response against nearby frames with tools compared to frames that are further away from the queries. This is because our per frame encodings may not be able to disentangle the presence and absence of tools over the varied anatomy present in our video sequences (Fig. 1). Direct evaluation achieves an AP of 71.56 in identifying frames with tools.

Table 1. Average Precision (AP) in frame-level detection of tool presence (* indicates supervised method)

Twinanda et al.*	52.5
Sahu et al.*	54.5
Raju et al.*	63.7
Jin et al.*	81.8
MMD-VAE	71.56
MMD-VAE + MCMC	73.93
MMD-VAE + FP	76.18

Approximate Inference. Since the encodings from the MMD–VAE show some separation, we expect a trained mixture of two Gaussians to capture the separation in the encodings and generalize to labeling our held-out test set. We found that the trained mixture model learned two clusters with distinct means and no overlap (Fig. 4). By treating each cluster as a binary tool indicator, we can predict labels for the encodings in test set. Compared to the direct evaluation method, we show improvement with an AP of 73.93. We hypothesize that our two assumptions that (1) the data can be represented as a mixture of two Gaussians, and (2) a sample belonging to a hidden configuration directly indicates tool presence or absence may be too strong and, therefore, limiting the improvement in AP. Instead, the clusters likely capture a combination of tool presence and anatomy variance, and precision could be further improved by either relaxing the assumptions or increasing the model complexity.

Future Prediction. We expect the addition of temporal information to allow for better disentanglement between tool motion from camera motion. We observe this improvement in the slightly greater separation in the encodings produced

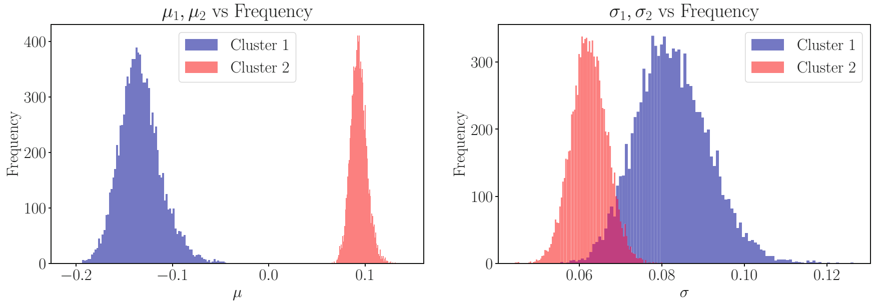


Fig. 4. Latent cluster parameter summaries for trained MMD-VAE + MCMC model with means (left) and standard deviations (right). The two configurations of C are described by $\mathcal{N}(-0.13, 0.0068)$ and $\mathcal{N}(0.093, 0.0039)$.

by MMD-VAE + FP than those produced without FP (Fig. 3b). This translates to further improvement in AP at 76.18. We hypothesize that larger gains in AP were again limited due to the short 5 frame sequence of encodings used to train the future prediction network. Using longer sequences may allow detection of tools over larger variations in anatomy and, therefore, improve overall results.

5 Conclusion and Future Work

We showed through our evaluations that it is possible to learn representations of endoscopic videos that allow us to identify surgical tool presence without supervision. We are able to detect frames containing tools directly from MMD–VAE encodings with an AP of 71.56. By performing approximate inference on these encodings, we are able to improve the AP of frame-level tool presence detection to 73.93. Finally, using the MMD–VAE encodings to perform future prediction allows us to further improve our AP to 76.18.

These results are comparable to those achieved by prior supervised methods evaluated on the M2cai16-tool dataset [29]. This dataset consists of 15 videos of cholecystectomy procedure, where each frame is labeled with the presence or absence of seven possible surgical tools in a multi-label fashion. As this work was evaluated on a different dataset, our immediate next step is to re-evaluate our unsupervised method on the M2cai16-tool dataset. This comparison will not only allow us to better understand how our methods compare against supervised methods, but also allow us to evaluate whether our methods can learn generalized representations across various surgical tools.

Going forward, we will explore whether variations in latent dimension and regularization for training our MMD-VAE can improve our ability to discriminate between frames with and without tools. We will also explore whether classification can be improved by accommodating the variance in anatomy by relaxing the assumption that our encodings are a mixture of two Gaussian states. Another space to explore will be whether larger sequences of video frame encodings allow us to better separate tool motion from camera motion. Although our initial work is on a limited endoscopic video dataset, our results are promising and our method can be easily applied to larger datasets with wider range of tools and anatomy since we do not rely on labels for training.

The ability to reliably identify frames containing tools can help the annotation process and can also enable further research in many different areas. For instance, methods that rely on endoscopic video frames without tools [17] can easily discard frames that are labeled as containing tools. Further, by treating features like optical flow vectors from sequences of frames with and without tools differently, we can work on identifying pixels containing tools without supervision. Unsupervised segmentation of tools, in turn, can enable unsupervised tool tracking and can have great impact on research toward video-based surgical activity recognition and skill assessment.

Acknowledgements. This work was supported by the Johns Hopkins University Provost's Postdoctoral fellowship, NVIDIA GPU grant, and other Johns Hopkins University internal funds. We would also like to thank Daniel Malinsky and Robert DiPietro for their invaluable feedback. We would also like to acknowledge the JHU Department of Computer Science providing a research GPU cluster.

References

1. Attia, M., Hossny, M., Nahavandi, S., Asadi, H.: Surgical tool segmentation using a hybrid deep CNN-RNN auto encoder-decoder. In: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 3373–3378, October 2017
2. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734 (2014)
3. Deng, J., Dong, W., Socher, R., Li, L., Kai, L., Li, F.-F.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255, June 2009. https://doi.org/10.1109/CVPR.2009.5206848
4. DiPietro, R., Hager, G.D.: Unsupervised learning for surgical motion by learning to predict the future. In: Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G. (eds.) MICCAI 2018. LNCS, vol. 11073, pp. 281–288. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00937-3_33
5. DiPietro, R., et al.: Recognizing surgical activities with recurrent neural networks. In: Medical Image Computing & Computer-Assisted Intervention, pp. 551–558 (2016)
6. Ephrat, M.: Acute sinusitis in HD (2013). www.youtube.com/watch?v=6niL7Poc_qQ
7. García-Peraza-Herrera, L.C., et al.: Real-time segmentation of non-rigid surgical tools based on deep learning and tracking. In: Computer-Assisted and Robotic Endoscopy (CARE), pp. 84–95 (2017)
8. Gers, F.A., Schmidhuber, J., Cummins, F.A.: Learning to forget: continual prediction with LSTM. Neural Comput. **12**, 2451–2471 (2000)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
10. Hoffman, M.D., Gelman, A.: The No-U-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. J. Mach. Learn. Res. **15**(1), 1593–1623 (2014)
11. Jin, A., Yeung, S., Jopling, J., Krause, J., Azagury, D., Milstein, A., Fei-Fei, L.: Tool detection and operative skill assessment in surgical videos using region-based convolutional neural networks. In: IEEE Winter Conference on Applications of Computer Vision (2018)
12. Karen Simonyan, A.Z.: Very deep convolutional networks for large-scale image recognition. ArXiv abs/1409.1556 (2014)
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv:1412.6980 (2014)
14. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. arXiv:1312.6114 (2013)

15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105. Curran Associates, Inc. (2012). http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
16. Lea, C., Vidal, R., Hager, G.D.: Learning convolutional action primitives for fine-grained action recognition. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 1642–1649, May 2016
17. Liu, X., et al.: Self-supervised learning for dense depth estimation in monocular endoscopy. In: Computer Assisted Robotic Endoscopy (CARE), pp. 128–138 (2018)
18. Malpani, A., Vedula, S.S., Chen, C.C.G., Hager, G.D.: A study of crowdsourced segment-level surgical skill assessment using pairwise rankings. Int. J. Comput. Assisted Radiol. Surg. **10**(9), 1435–1447 (2015). https://doi.org/10.1007/s11548-015-1238-6
19. Murphy, K.P.: Machine Learning: A Probabilistic Perspective. MIT Press, Cambridge (2012)
20. Pakhomov, D., Premachandran, V., Allan, M., Azizian, M., Navab, N.: Deep Residual Learning for Instrument Segmentation in Robotic Surgery. arXiv:1703.08580 (2017)
21. Paszke, A., et al.: Automatic differentiation in pytorch. In: NIPS-W (2017)
22. Raju, A., Wang, S., Huang, J.: M2cai surgical tool detection challenge report (2016)
23. Sahu, M., Mukhopadhyay, A., Szengel, A., Zachow, S.: Tool and phase recognition using contextual CNN features. ArXiv abs/1610.08854 (2016)
24. Shvets, A.A., Rakhlin, A., Kalinin, A.A., Iglovikov, V.I.: Automatic instrument segmentation in robot-assisted surgery using deep learning. In: 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 624–628 (2018)
25. Srivastava, N., Mansimov, E., Salakhutdinov, R.: Unsupervised learning of video representations using LSTMS. In: Proceedings 32nd International Conference on International Conference on Machine Learning. ICML 2015, vol. 37, pp. 843–852. JMLR.org (2015)
26. Stan Development Team: PyStan: the Python interface to Stan, Version 2.17.1.0. (2018). http://mc-stan.org
27. Szegedy, C., et al.: Going deeper with convolutions. In: Computer Vision and Pattern Recognition (CVPR) (2015). http://arxiv.org/abs/1409.4842
28. Tsui, C., Klein, R., Garabrant, M.: Minimally invasive surgery: national trends in adoption and future directions for hospital strategy. Surgical Endoscopy **27**(7), 2253–2257 (2013)
29. 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. Imag. **36**, 86–97 (2016)
30. Zhao, S., Song, J., Ermon, S.: InfoVAE: Information Maximizing Variational Autoencoders. arXiv:1706.02262 (2017)
31. Zhu, M.: Recall, precision and average precision. In: Department of Statistics and Actuarial Science, University of Waterloo, Waterloo **2**, p. 30 (2004)