Surgical Activity Recognition in Robot-Assisted Radical Prostatectomy Using Deep Learning

Aneeq Zia[1](✉), Andrew Hung[2], Irfan Essa[1], and Anthony Jarc[3]

[1] Georgia Institute of Technology, Atlanta, GA, USA
aneeqzia@gatech.edu
[2] University of Southern California, Los Angeles, CA, USA
[3] Medical Research, Intuitive Surgical Inc., Norcross, GA, USA

Abstract. Adverse surgical outcomes are costly to patients and hospitals. Approaches to benchmark surgical care are often limited to gross measures across the entire procedure despite the performance of particular tasks being largely responsible for undesirable outcomes. In order to produce metrics from tasks as opposed to the whole procedure, methods to recognize automatically individual surgical tasks are needed. In this paper, we propose several approaches to recognize surgical activities in robot-assisted minimally invasive surgery using deep learning. We collected a clinical dataset of 100 robot-assisted radical prostatectomies (RARP) with 12 tasks each and propose *'RP-Net'*, a modified version of InceptionV3 model, for image based surgical activity recognition. We achieve an average precision of 80.9% and average recall of 76.7% across all tasks using RP-Net which out-performs all other RNN and CNN based models explored in this paper. Our results suggest that automatic surgical activity recognition during RARP is feasible and can be the foundation for advanced analytics.

1 Introduction

Adverse outcomes are costly to the patient, hospital, and surgeon. Although many factors contribute to adverse outcomes, the technical skills of surgeons are one important and addressable factor. Virtual reality simulation has played a crucial role to train and improve the technical skills of surgeons, however, intraoperative assessment has been limited to feedback from attendings and/or proctors. Aside from the qualitative feedback from experienced surgeons, quantitative feedback has remained abstract to the level of an entire procedure, such as total duration. Performance feedback for one particular task within a procedure might be more helpful to direct opportunities of improvement. Similarly, statistics from the entire surgery may not be ideal to show an impact on outcomes. For example, one might want to closely examine the performance of a single task if certain adverse outcomes are related to only that specific step of the entire procedure [1]. Scalable methods to recognize automatically when

A. F. Frangi et al. (Eds.): MICCAI 2018, LNCS 11073, pp. 273–280, 2018.
https://doi.org/10.1007/978-3-030-00937-3_32

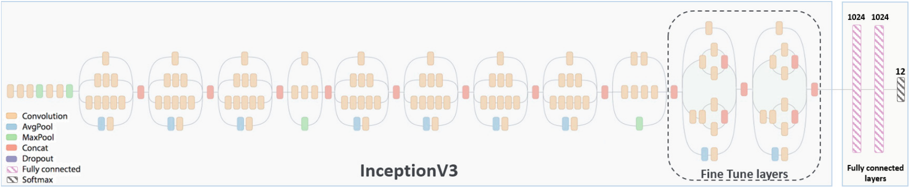

Fig. 1. RP-Net architecture. The portion shown in blue is the same as InceptionV3 architecture, whereas the green portion shows the fully connected (fc) layers we add to produce RP-Net. The number of units for each fc layers is also shown. Note the last two layers of InceptionV3 are fine-tuned in RP-Net.

particular tasks occur within a procedure are needed to generate these metrics to then provide feedback to surgeons or correlate to outcomes.

The problem of surgical activity recognition has been of interest to many researchers. Several methods have been proposed to develop algorithms that automatically recognize the phase of surgery. For laparoscopic surgeries, [2] proposed *'Endo-Net'* for recognizing surgical tools and phases in cholecystectomy using endoscopic images. In [3], RNN models were used to recognize surgical gestures and maneuvers using kinematics data. In [4], unsupervised clustering methods were used to segment training activities on a porcine model. In [5], hidden markov models were used to segment surgical workflow within laparoscopic cholecystectomy.

In this work, we developed models to detect automatically the individual steps of robot-assisted radical prostatectomies (RARP). Our models break a RARP into its individual steps, which will enable us to provide tailored feedback to residents and fellows completing only a portion of a procedure and to produce task-specific efficiency metrics to correlate to certain outcomes. By examining real-world, clinical RARP data, this work builds foundational technology that can readily translate to have direct clinical impact.

Our contributions are, (1) a detailed comparison of various deep learning models using image and robot-assisted surgical system data from clinical robot-assisted radical prostatectomies; (2) RP-Net, a modified InceptionV3 architecture that achieved the highest surgical activity recognition performance out of all models tested; (3) a simple median filter based post processing step for significantly improving procedure segmentation accuracies of different models.

2 Methodology

The rich amount of data that can be collected from the da Vinci (dV) surgical system (Intuitive Surgical, Inc., Sunnyvale, CA USA) enables multiple ways to explore recognition of the type of surgical tasks being performed during a procedure. Our development pipeline involves the following steps: (1) extraction of endoscopic video and dV surgical system data (kinematics and a subset of events), (2) design of deep learning based models for surgical task recognition, and (3) design of post-processing models to filter the initial procedure segmentation output to improve performance. We provide details on modeling below and on our dataset in the next section.

System Data Based Models: The kind of hand and instrument movements surgeons make during procedures can be very indicative of what types of task they are performing. For example, a dissection task might involve static retraction and blunt dissection through in and out trajectories, whereas a suturing task might involve a lot of curved trajectories. Therefore, models that extract motion and event based features from dV surgical system data seem appropriate for task/activity recognition. We explore multiple Recurrent Neural Network (RNN) models using only system data given the recent success of RNNs to incorporate temporal sequences. Since there are multiple data streams coming from the dV surgical system, we employ two types of RNN architectures - *single stream* (SS) and *multi-stream* (MS). For SS, all data streams are concatenated together before feeding them into a RNN. Whereas, for MS, each data stream is fed into individual RNNs after which the outputs of each RNN are merged together using a fully-connected layer to produce predictions. For training both architecture types, we divide our procedure data into windows of length W. At test time, individual windows of the procedure are classified to produce the output segmentation.

Video Based Models: Apart from the kind of motions a surgeon makes, a lot of task representative information is available in the endoscopic video stream. Tasks which are in the beginning could generally look more *'yellow'* due to the fatty tissues, whereas tasks during the later part of the surgery could look much more *'red'* due to the presence of blood after dissection steps. Moreover, the type and relative location of tools present in the image can also be very indicative of the step that the surgeon is performing. Therefore, we employ various image based convolutional neural networks (CNN) for recognizing surgical activity using video data. Within the CNNs domain, there are two type of CNN architectures that are popular and have been proved to work well for the purpose of recognition. The first type uses single images only with two-dimensional (2D) convolutions in the CNN architectures. Examples of such networks include VGG [6], ResNet [7] and InceptionV3 [8]. The second type of architecture uses a volume of images as input (e.g., 16 consecutive frames from the video) and employs three-dimensional (3D) convolutions instead of 2D. C3D is an example of such model [9]. A potential advantage of 3D models is that they can learn spatio-temporal features from video data instead of just spatial features. However, this comes at the cost of requiring more data to train as well as longer overall training times. For our task of surgical activity recognition, we employ both types of CNN models and also propose *'RP-Net'* (Radial Prostatectomy Net), which is a modified version of InceptionV3 as shown in Fig. 1.

Post-processing: Since there are parts of various tasks that are very similar visually and in terms of motions the surgeon is making, the predicted procedure segmentation can have *'spikes'* of mis-classifications. However, it can be assumed that the predicted labels would be consistent within a small window. Therefore,

Table 1. Dataset: the 12 steps of robot-assisted radical prostatectomy and general statistics.

Task no.	Task name	Mean time (sec)	Number of samples
T1	Mobilize colon/drop bladder	1063.2	100
T2	Endopelvic fascia	764.2	98
T3	Anterior bladder neck dissection	164.9	98
T4	Posterior bladder neck dissection	617.5	100
T5	Seminal vesicles	686.8	100
T6	Posterior plane/denonvilliers	171.2	99
T7	Predicles/nerve sparing	510.6	100
T8	Apical dissection	401.1	100
T9	Posterior anastomosis	403.1	100
T10	Anterior anastomosis	539.7	100
T11	Lymph node dissection left	999.6	100
T12	Lymph node dissection right	1103.6	100

in order to remove such noise from the output, we employ a simple running window median filter of length F as a post-processing step. For corner cases, we append the start and end of the predicted sequence with the median of first and last window of length F, respectively, in order to avoid mis-classifications of the corner cases by appending zeros.

3 Experimental Evaluation

Dataset: Our dataset consisted of 100 robot-assisted radical prostatectomies (RP) completed at an academic hospital. The majority of procedures were completed by a combination of residents, fellows, and attending surgeons. Each RP was broken into approximately 12 standardized tasks. The order of these 12 tasks varied slightly based on surgeon preference. The steps of each RP were annotated by one resident. A total of 1195 individual tasks were used. Table 1 shows general statistics of our dataset.

Each RP recording included one channel of endoscopic video, dV surgical system kinematic data (e.g., joint angles, endpoint pose) collected at 50 Hz, and dV surgical system event data (e.g., camera movement start/stop, energy application on/off).

The dV surgical system kinematic data originated from the surgeon console (SSC) and the patient side cart (SI). For both the SSC and SI, the joint angles for each manipulandum and the endpoint pose of the hand controller or instrument were used. In total, there were 80 feature dimensions for SSC and 90 feature dimensions for SI. The dV surgical system event data (EVT) consisted of many events relating to surgeon interactions with the dV surgical system originating at the SSC or SI. In total, there were 87 feature dimensions for EVT.

Data Preparation: Several pre-processing steps were implemented. The endoscopic video was downsampled to 1 frame per second (fps) resulting in 1.4 million images in total. Image resizing and rescaling was model specific. All kinematic data was downsampled by a factor of 10 (from 50 Hz to 5 Hz). Different window lengths (in terms of the number of samples) W (50, 100, 200 and 300) were tried for training the models and $W = 200$ performed the best. We used zero overlap when selecting windows for both training and testing. Mean normalization was applied to all feature dimensions for the kinematic data. All events from the dV surgical system data that occured within each window W were used as input for to our models. The events were represented as a unique integers with corresponding timestamps.

Model Training and Parameter Selection: For RNN based models, we implemented both SS and MS architectures for all possible combinations of the three data streams (SSC, SI, and EVT). Estimation of model hyperparameters was done via a grid search on the number of hidden layers (1 or 2), type of RNN unit (Vanilla, GRU or LSTM), number of hidden units per layer (8, 16, 32, 64, 128, 256, 512 or 1024) and what dropout ratio to use (0, 0.2 or 0.5). For each parameter set, we also compared forward and bi-directional RNN. The best performances were achieved using single layered bi-directional RNNs with 256 LSTM units and a dropout ratio of 0.2. Hence, all RNN based results presented were evaluated using these parameters for SS and MS architecture types.

In CNN based models, we used two approaches - training the networks from randomly initialized weights and fine-tuning the networks from pre-trained weights. For all models, we found that fine-tuning was much faster and achieved better accuracies. For single image based models, we used ImageNet [10] pre-trained weights while for C3D we used Sports-1M [11] pretrained weights. We found that fine-tuning several of the last convolutional layers led to the best performances across models. For the proposed RP-Net, the last two convolutional modules were fine-tuned (as shown in Fig. 1) and the last fully connected layers were trained from random initialization.

For both RNN- and CNN-based models, the dataset was split to include 70 procedures for training, 10 procedures for validation, and 20 procedures for test.

For the post-processing step, we evaluated performances of all models for values of F (median filter length) ranging from 3 to 2001, and choose a window length that led to maximum increase in model performance across different methods. The final value of F was set to 301. All parameters were selected based on the validation accuracy.

Evaluation Metrics: For a given series of ground truth labels $G \in \Re^N$ and predictions $P \in \Re^N$, where N is the length of a procedure, we evaluate multiple metrics for comparing the performance of various models. These include average precision (AP), average recall (AR) and Jaccard index. Precision is evaluated using $P = \frac{tp}{tp+fp}$, recall using $R = \frac{tp}{tp+fn}$ and Jaccard index using $J = \frac{tp}{tp+fp+fn}$, where tp, fp and fn represent the true positives, false positives and false negatives, respectively.

Table 2. Surgical procedure segmentation results using different models. Each cell shows the average metric values across all procedures and tasks in the test set with standard deviations using the original predictions and filtered predictions in the form *original* | *filtered*. For LSTM models, the modalities used are given in square brackets while the architecture type used is given in parentheses.

Model Type	Average Precision	Average Recall	Average Jaccard Index
LSTM [ssc+si] (MS)	0.585±0.19 | 0.595±0.21	0.565±0.21 | 0.572±0.21	0.629±0.18 | 0.645±0.19
LSTM[ssc+si] (SS)	0.559±0.14 | 0.578±0.15	0.526±0.16 | 0.551±0.16	0.582±0.16 | 0.606±0.17
LSTM[ssc+evt] (MS)	0.625±0.13 | 0.648±0.13	0.572±0.16 | 0.593±0.17	0.633±0.18 | 0.662±0.19
LSTM[ssc+evt] (SS)	0.625±0.13 | 0.641±0.13	0.567±0.21 | 0.593±0.22	0.625±0.18 | 0.651±0.19
LSTM[ssc+si+evt] (MS)	0.437±0.29 | 0.458±0.31	0.226±0.31 | 0.471±0.32	0.552±0.15 | 0.582±0.16
LSTM[ssc+si+evt] (SS)	0.544±0.13 | 0.579±0.12	0.518±0.17 | 0.546±0.17	0.575±0.15 | 0.603±0.17
InceptionV3	0.662±0.12 | 0.782±0.14	0.642±0.15 | 0.759±0.17	0.666±0.07 | 0.786 ±0.08
VGG-19	0.549±0.16 | 0.695±0.19	0.481±0.2 | 0.573±0.22	0.529±0.08 | 0.634±0.11
ResNet	0.621±0.1 | 0.713±0.12	0.582±0.21 | 0.673±0.25	0.622±0.07 | 0.728±0.08
C3D	0.442±0.17 | 0.352±0.21	0.417±0.19 | 0.367±0.24	0.504±0.06 | 0.418±0.12
RP-Net	**0.714±0.12 | 0.809±0.13**	**0.676±0.2 | 0.767±0.23**	**0.700±0.05 | 0.808±0.07**

4 Results and Discussion

The evaluation metrics for all models are shown in Table 2. RP-Net achieved the highest scores across all evaluation metrics out of all models (see last row in Table 2). In general, we observed that the image-based CNN models (except for C3D) performed better than the RNN models. Within LSTM models, MS architecture performed slightly better than SS with the SSC+EVT combination achieving the best performance. For nearly all models, post-processing significantly improved task recognition performance.

Figure 2 shows the confusion matrix of RP-Net with post-processing. The model performed well for almost all the tasks individually except for task 9. However, we can see that most of the task 9 samples were classified as task 10. Tasks 9 and 10 are very related - they are two parts of one overall task (posterior and anterior anastomosis). Furthermore, the images from these two tasks were quite similar given they show anatomy during reconstruction after extensive dissection and energy application. Hence, one would expect that the model could be confused on these two tasks. This is also the case for tasks 3 and 4 - anterior and posterior bladder neck dissection, respectively.

Figure 3 shows several visualizations of the segmentation results as color-coded bars. Undesired spikes in the predicted surgical phase were present when using the output of RP-Net directly. This can be explained by the fact that the model has no temporal information and classifies only using a single image which can lead to mis-classifications since different tasks can look similar at certain points in time. However, using the proposed median filter for post-processing significantly remove such noise and produces a more consistent output (compare middle to bottom bars for all three sample segmentation outputs in Fig. 3).

Despite not having temporal motion information, single image-based models recognize surgical tasks quite well. One reason for this result could be due to the significantly large dataset available for single-image based models. Given the presented RNN and C3D models use a window from the overall task as input,

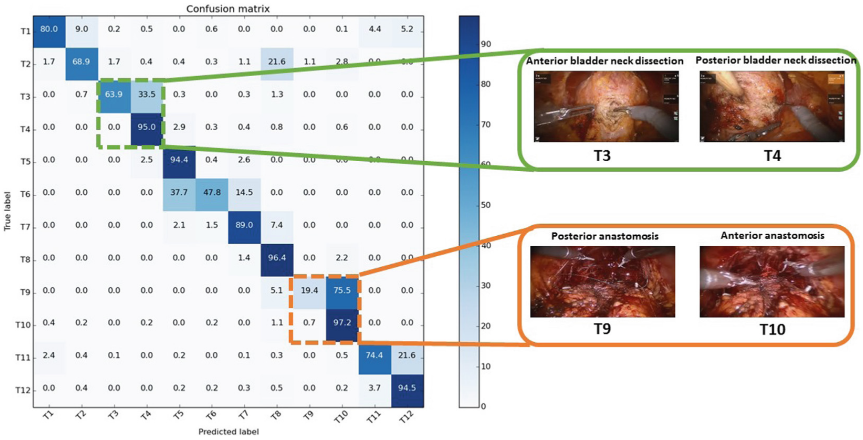

Fig. 2. Confusion matrix of results using RP-Net with post-processing. Sample images of tasks between which there is a lot of *'confusion'* are also shown.

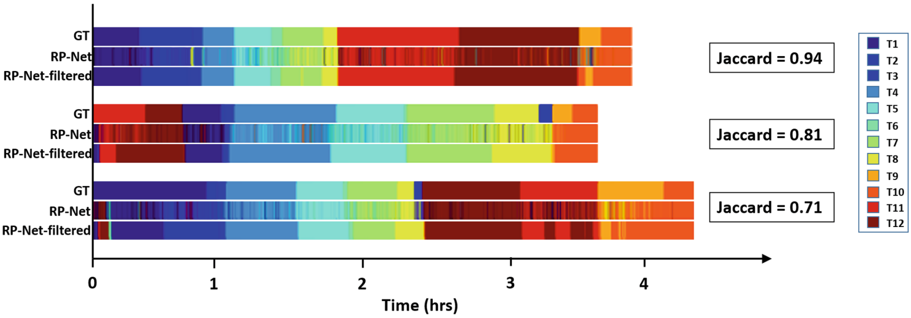

Fig. 3. Sample segmentation outputs for the best, median and lowest jaccard index achieved (from top to bottom, respectively). Within each plot, the top bar denotes the ground truth, the middle one shows the output of RP-Net, while the lowest one shows the output after applying the median filter. Please see Table 1 for task names.

the amount of training data available for such models reduces by a factor of the length of window segment. Additionally, the RNN models might not have performed as well as similar work because in this work we recognized gross tasks directly whereas prior work focused on sub-task gestures and/or maneuvers [3]. Finally, C3D models remain difficult to train. Improved training of these models could lead to better results, which aligns with the intuition that temporal windows of image frames could provide relevant information for activity recognition.

5 Conclusion

In this paper, we proposed a deep learning model called RP-Net to recognize the steps of robot-assisted radical prostatectomy (RARP). We used a clinically-relevant dataset of 100 RARPs from one academic center which enables translation of our models to directly impact real-world surgeon training and medical research. In general, we showed that image-based models outperformed models using only surgeon motion and event data. In future work, we plan to develop novel models that optimally combine motion and image features while using larger dataset and to explore how our models developed for RARP extend to other robot-assisted surgical procedures.

References

1. Hung, A.J., et al.: Utilizing machine learning and automated performance metrics to evaluate robot-assisted radical prostatectomy performance and predict outcomes. J. Endourol. **32**(5), 438–444 (2018)
2. 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)
3. DiPietro, R., et al.: Recognizing surgical activities with recurrent neural networks. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9900, pp. 551–558. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46720-7_64
4. Zia, A., Zhang, C., Xiong, X., Jarc, A.M.: Temporal clustering of surgical activities in robot-assisted surgery. Int. J. Comput. Assist. Radiol. Surg. **12**(7), 1171–1178 (2017)
5. 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)
6. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
8. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
9. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 4489–4497. IEEE (2015)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: alarge-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 248–255. IEEE (2009)
11. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1725–1732 (2014)