Surgeon Technical Skill Assessment using Computer Vision based Analysis

Hei Law HEILAW@UMICH.EDU
Computer Science and Engineering
University of Michigan
Ann Arbor, Michigan, USA

Khurshid Ghani KGHANI@MED.UMICH.EDU
Department of Urology
University of Michigan
Ann Arbor, Michigan, USA

Jia Deng JIADENG@UMICH.EDU
Computer Science and Engineering
University of Michigan
Ann Arbor, Michigan, USA

Abstract

In this paper, we propose a computer vision based method to assess the technical skill level of surgeons by analyzing the movement of robotic instruments in robotic surgical videos. First, our method leverages the power of crowd workers on the internet to obtain high quality data in a scalable and cost-efficient way. Second, we utilize the high quality data to train an accurate and efficient robotic instrument tracker based on the state-of-the-art Hourglass Networks. Third, we assess the movement of the robotic instruments and automatically classify the technical level of a surgeon with a linear classifier, using peer evaluations of skill as the reference standard. Since the proposed method relies only on video data, this method has the potential to be transferred to other minimally invasive surgical procedures.

1. Introduction

Robotic surgery is a widely adopted minimally invasive approach to surgery. In the United States, robotic prostatectomy is the most commonly performed operation for patients with prostate cancer (Leow et al., 2016). However, patients undergoing robotic prostatectomy demonstrate a wide range in patient outcomes which have lead many surgeons to believe that a surgeon's technical skill is a pivotal determinant of patient outcomes following surgery. There is therefore a need for clinically relevant, robust, and practical strategies for assessing and ultimately improving the technical proficiency of surgeons performing robotic prostatectomy. Conventionally, technical skill assessment of surgeons is done by peer surgeon review, which is a time consuming, costly and non-scalable process (Ghani et al., 2016). As robotic surgical procedures can be easily recorded as video files, we now have an opportunity to analyze robotic surgical procedures for surgical skill and quality by leveraging the power of computers.

Surgeons are trained to concentrate on their hand or instrument movements during surgery and prior work (Ghani et al., 2016) shows that lay people with no knowledge of surgery can also differentiate high and low skill surgeons based on reviewing robotic surgery

videos. To assess the technical skill of a surgeon performing robotic surgery, it is important to evaluate the proficiency of a surgeon in using the robotic instruments throughout an operation. Hence, when assessing videos, we must be able to track the robotic instruments and capture the motion information of the instruments. However, tracking a surgical instrument is a very challenging computer vision task due to occlusion of the target and interaction between tissue and other instruments. Thanks to recent advances in crowdsourcing, computer vision and machine learning algorithms, it is now possible to design a computer algorithm to track an object efficiently and accurately.

Recently, convolution neural networks (ConvNet) (LeCun et al., 1998) have become a dominant approach in many different computer vision tasks. ConvNet is an artificial neural network which mimics the human vision system and was first proposed in the 1990s. Its remarkable ability in learning high-level concept from data was not unveiled until recent years. Because of the availability of the large-scale datasets, ConvNets now demonstrate state-of-the-art performance in many computer vision tasks, outperforming many approaches which use hand-crafted features. However, training a ConvNet requires a large-scale of high quality training data. Hence, crowdsourcing provides us with a cheap and scalable way to collect large amounts of high quality data for training the ConvNet. Recently, lots of large-scale datasets (Deng et al., 2009; Lin et al., 2014) designed for different computer vision tasks have been annotated by crowdsourcing. For example, ImageNet (Deng et al., 2009) is one of the largest crowdsourced datasets and consists of more than a million images for image classification task. Since ImageNet 2012, ConvNets (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014b; He et al., 2016) have demonstrated state-of-the-art performance in the image classification task. Later, ConvNets have been applied to many other computer vision tasks, such as object detection (Ren et al., 2015), semantic segmentation (Long et al., 2015) and human pose estimation (Newell et al., 2016).

Therefore, in this paper, we introduce a computer vision-based analysis, which leverages crowdsourcing, ConvNets and advanced machine learning algorithms, to analyze surgical videos, and assess the technical skill level of surgeons performing robotic prostatectomy. Although we focus on assessing the technical skill of surgeons performing robotic prostatectomy, our method may be transferred to any minimally invasive surgical procedure that can be recorded, and has the potential to determine surgical skill assessment in a variety of scenarios as the proposed method relies only on video data.

2. Cohort

This study is conducted as a collaboration between surgeons within the Michigan Urological Surgical Improvement Collaborative (MUSIC). The statewide quality improvement consortium consists of over 40 urology practices and is aimed at improving prostate cancer care. All surgeons performing robotic prostatectomy were invited to take part in a video review initiative, by submitting an unedited complete video of a robotic prostatectomy that they had performed. Videos were processed at a Coordinating Center, where each video was stripped of all patient and surgeon identifiers, and edited to multiple parts of the case. For this study we analyzed the part of the operation where the bladder is reattached to the urethra after the prostate is excised: the vesico-urethral anastomosis. We used 12 videos from 12 different surgeons performing the anastomosis; video duration ranged from 9 to 36 minutes. The resolution of the videos is 720×480. The video-clips for each surgeon were

then placed on a secure video platform within a registry, where peer surgeons were able to evaluate the videos in a blinded manner for technical skill.

Peer assessment of the technical skill of surgeons was performed using a validated instrument: Global Evaluative Assessment of Robotic Skills (GEARS) (Goh et al., 2012). We use a modified GEARS to assess global robotic skill which includes domains of (1) efficiency, (2) force sensitivity, (3) bimanual dexterity, (4) depth perception and (5) robotic control scored on a 1-5 Likert scale with a maximum total score of 25 (minimum 5). Evaluation of individual video clips was performed by peer surgeon reviewers.

3. Methods

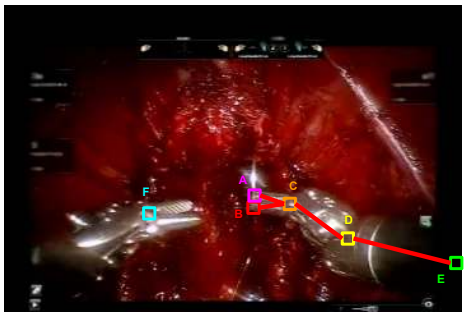

Figure 1: We used five keypoints of the right sided robotic instrument (needle driver) including two tips (A, B), apex (C), joint (D) and the arm endpoint (E). We also tracked the apex (F) of the left sided needle driver.

Unlike instruments used for conventional laparoscopic surgery, robotic surgery instruments are articulated instruments that have six degrees of freedom, with movements almost like the human hand. To capture the motion of robotic instruments, we track six keypoints of two robotic instruments (robotic needle drivers), including the tips, apex, joint and instrument arm endpoint of the right-sided robotic instrument as well as the apex of the left-sided robotic instrument. All the keypoints are shown in Fig. 1.

Our computer vision based analysis can be divided into three steps including crowdsourcing, robotic instrument tracking, and technical skill assessment. First, to construct a large scale dataset for training our ConvNet, we first annotate all the keypoints of the robotic instruments in our videos by crowdsourcing. Second, we train a ConvNet to learn to detect and track the keypoints using the crowdsourced annotations. Finally, we train a support vector machine (SVM) to classify the skill of a surgeon using the tracking results.

3.1 Crowdsourcing

We annotate the keypoints of the robotic instruments in all the videos by crowdsourcing on Amazon Mechanical Turk (AMT). We build our web-based system upon a video annotation system, VATIC, proposed by Vondrick et al. (2013). VATIC is an AMT crowdsourcing system designed to build large-scale video dataset for computer vision researches. We design a line annotation interface to help workers annotate multiple keypoints of a robotic instrument easily in a frame as shown in Fig. 2. Instead of annotating the keypoints by

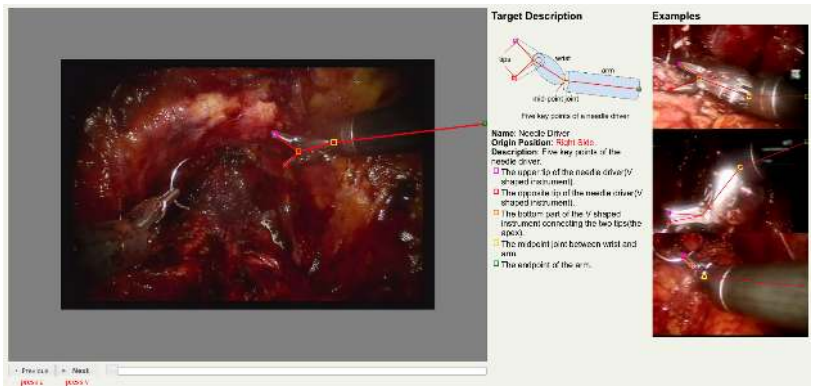

Figure 2: User interface of our crowdsourcing system on Amazon Mechanical Turk. Crowd workers annotate the keypoints of the robotic instruments by dragging the small square boxes. We also provide some examples in our interface to help workers identify and locate each keypoint.

clicking on the regions of interest, the workers indicate the locations of keypoints by dragging the boxes on the line. This line interface allows the crowd workers to annotate the keypoints more accurately and easily. If the instrument is not visible in the frame, the workers are instructed to place the line annotation outside of the frame.

For each video, we divide it into more manageable smaller sized chunks. Each chunk of a video consists of 20 consecutive frames and does not overlap with other jobs. Each chunk is treated as a *regular* job and we need one worker to work on it. Since the frames within each chunk are consecutive, workers can annotate every frame easily once they annotate the first frame of a chunk. To maintain the quality of the annotations , we introduce *groundtruth* jobs, which are used in evaluating the quality of the *regular* jobs. A *groundtruth* job is created by randomly extracting 3 frames from each *regular* job, which are then stacked together to create a chunk for annotation. Each *groundtruth* job consists of 20 discontinuous frames. Multiple crowd workers are assigned to work on a *groundtruth* job and the average of their annotations is used as the value. Following the completion of *groundtruth* jobs, when a crowd worker submits their annotations for *regular* jobs to our server, our server automatically evaluates the quality of their annotation against annotations from the *groundtruth* jobs. If a crowd worker has at least 2 out of 3 frames within a threshold of accuracy, these annotations are accepted and the crowd worker is paid. Otherwise, the annotations are rejected and the job is resubmitted for other workers to complete. If a job is rejected more than 3 times, it will be discarded.

3.2 Robotic Instruments Tracking by Hourglass Networks

Recently, Newell et al. (2016) proposed stacked hourglass networks for the task of human pose estimation, where the goal is to detect human joints in an image. Their hourglass network achieves state-of-the-art results in the challenging human pose estimation datasets (Sapp and Taskar, 2013; Andriluka et al., 2014). A stacked hourglass network consists of multiple hourglass modules. Each hourglass module can capture both global and local information within a single unified structure through a series of downsampling layers, upsampling layers and skip layers. Please refer to Newell et al. (2016) for more details.

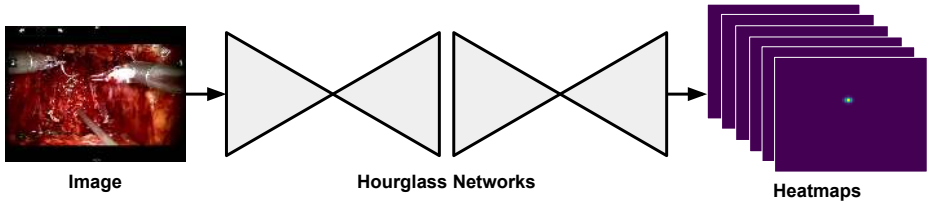

Figure 3: Architecture of the stacked hourglass network used for tracking key points of robotic instruments. We stack two hourglass modules in our network. We use the full image as the input to the network. The network predicts a heatmap for each key points.

Since our task shares lots of similarity between human pose estimation task, we adopt the stacked hourglass network to detect the keypoints of the robotic instruments. The network learns to predict the location of each keypoint of the robotic instruments. As shown in Fig. 3, we resize the image to 255×255 as the input to the network. For each keypoint of the robotic instruments, the network generates a heatmap of size 64×64 indicating the keypoint location. Hence, the network predicts 6 heatmaps for a single image. We can obtain the location of each keypoint by thresholding the corresponding heatmap.

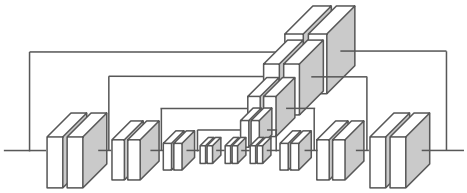

Figure 4: Architecture of the hourglass module used in our stacked hourglass network. Each rectangle represents a conv-bn-relu module.

We use two hourglass modules in our stacked hourglass network. Before the hourglass modules, we apply a 7×7 convolutional layer, a max pooling layer and several 3×3 convolutional layers to downsample the input image to one forth of its original size and pre-process it. We design the hourglass module following the architecture in Newell and Deng (2016), which replaces the residual module with a simple module consisting of a convolutional layer, a batch normalization layer and a ReLU layer. In our hourglass module, there are two 3×3 convolutional layers with 128 features before each max pooling or upsampling layers and two 3×3 convolutional layers in the middle of the module. The architecture of our hourglass module is shown Fig. 4. Each hourglass module generates a set of heatmaps indicating the locations of keypoints. As suggested in Newell et al. (2016), we also apply loss function to the intermediate outputs to provide intermediate supervision when we train the network. After training the network, we use only the output of the last hourglass module as the final prediction. During training, we indicate the location of each keypoint using a 5×5 Gaussian distribution and use the mean squared error as the objective function.

We adopt Adam (Kingma and Ba, 2014) to optimize our loss function. When we train the network, we use a batch size of 64 and we train it for 20000 iterations using a learning

rate of 0.001 with random initialization. The input images are resized to 255×255 and the output heatmaps are of size 64×64. To avoid overfitting, we perform data augmentation on our input images. We adjust the brightness, contrast, saturation of an image. We also apply fancy PCA (Krizhevsky et al., 2012) using eigenvalues and eigenvectors of a RGB covariance matrix of a random subset of the training images. Finally, we whiten the image with mean and standard deviation of RGB values of random training images. We implement our network using PyTorch. During testing, we resize images to 255×255 and do whitening using the same mean and standard deviation.

3.3 Technical Skill Classification with Support Vector Machine

Given the keypoints of the robotic instruments within the surgical videos, we classify the technical skill level of a surgeon by applying a support vector machine (SVM).

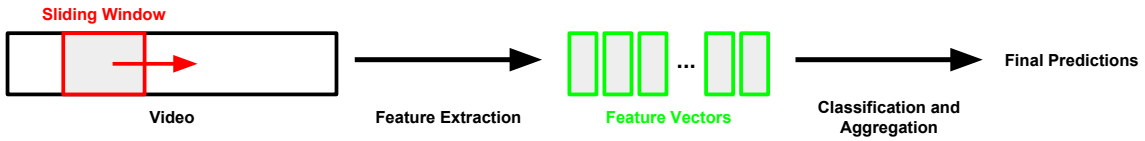

Figure 5: We compute a feature vector describing the movement of robotic instruments for each window. Then, we classify each window and aggregate classification information to make a final prediction. This figure is best shown in color.

Inspired by Simonyan and Zisserman (2014a) work on action recognition in video, when we classify the skill level of a surgeon, we divide the video into many small segments by using a sliding window approach. Each window consists of k continuous frames of a video. For each window, we compute a feature vector describing the motion of the robotic instruments within the window as shown in Fig. 5. We evaluate how likely this window belongs to a surgeon with higher skill level by applying an SVM. We calculate the scores for a random subset of the sliding windows and make a final prediction by averaging the scores from all windows. This allows us to capture details of the movement of robotic instruments with a short time period and aggregate information from different periods of the full video.

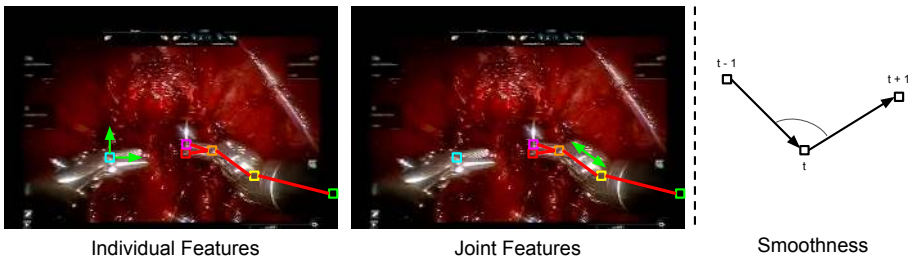

Figure 6: The features are computed both individually and jointly. The individual features describe the motion of each keypoints independently. Meanwhile, the joint features describe the relationships between different keypoints in the robotic instruments. The smoothness is defined to be whether the angle between two consecutive movement is greater than $\frac{\pi}{3}$.

The feature vector of each window consists of both individual features and joint features obtained from the keypoints of the robotic instruments as shown in Fig. 6. The individual features are computed independently for each keypoint in the robotic instruments. The individual features include the average velocity, average acceleration (change of velocity), average smoothness and the length of the trajectory of each keypoints within a small window. A keypoint is considered as moving smoothly if the angle between three consecutive frames is larger than $\frac{\pi}{3}$ as illustrated in Fig. 6. While the individual features describe the motion of each keypoint independently, the joint features describe the relationships between different keypoints. For the joint features, we consider the distance and the change of distance between different pairs of keypoints including tip and apex, apex and joint, as well as joint and end point. Furthermore, we include the angle and the change of angle between two tips in the joint features.

4. Results

4.1 Evaluation Approach/Study Design

	A	B	C	D	E	F	G	H	I	J	K	L	Variance
Depth	3.62	4	4.38	4.44	3.23	3.15	3.77	3.75	3.93	3.5	4.18	4.25	0.181
Bi-manual	3.38	4.27	3.88	4.78	3.08	3.25	3.62	3.5	3.93	3.25	4.36	3.88	**0.267**
Efficiency	2.91	4.09	4.13	4.67	2.92	2.55	3.31	3.25	3.87	3.13	4.18	3.75	**0.4136**
Force	4.00	3.45	4.25	4.33	3.23	3.35	3.54	3.5	.387	3.25	4.36	4.63	0.240
Robotic	3.77	4	4.5	4.89	3.54	3.45	3.85	4	4.2	3.75	4.36	3.88	0.171
Average	3.53	3.96	4.23	4.62	3.2	3.15	3.62	3.6	3.96	3.38	4.29	4.08	0.212
Skill Level	lower	higher	higher	higher	lower	lower	higher	higher	higher	higher	higher	higher	

Table 1: Our dataset consists of 12 videos, which are all assessed by peer surgeons. Among all the metrics, "Bi-manual dexterity" and "Efficiency" have the highest variance of the scores. Hence, we divide the surgeons into two skill groups, lower skill and higher skill, by using thresholds for scores in "Bi-manual dexterity" (i.e. < 3.5) and "Efficiency" (i.e. < 3) scores. Surgeon A, E and F are considered with lower skill level, while the remaining surgeons are considered with higher skill level.

Our dataset consists of 12 videos which are peer assessed by other surgeons. Peer surgeons evaluate the skill level of a surgeon using 5 metrics, including "Depth Perception" (Depth), "Bimanual Dexterity" (Bi-manual), "Efficiency" (Efficiency), "Force Sensitivity" (Force) and "Robotic Control" (Robotic). The average scores of each metric are shown in Tab. 1. Since all the surgeons performing prostatectomy in our dataset are skilled surgeons, the range in the average scores is narrow. However, we observe two metrics, "Bimanual Dexterity" and "Efficiency", which have higher variance than other metrics. "Bimanual Dexterity" measures how well a surgeon uses both hand in the surgery, meanwhile "Efficiency" measures how well a surgeon executes the movement. Hence, we divide the surgeons into two groups, lower skill and higher skill, based on these two metrics. The goal of our task is to then automatically track the instruments of the videos and classify the skill level of the surgeons. Since we only have 12 videos in total, to accurately evaluate the performance of our method, we adopt leave-one-out cross validation in all our subsequent experiments. We train our neural network and SVM on 11 videos and test on 1 video. We then repeat this procedure 12 times.

4.2 Crowdsourcing

We apply our system to collect keypoint annotations for 12 videos, which consists of 146,309 frames in total. Including the jobs we resubmitted, we submitted 8274 jobs, of which only 1339 (around 16%) jobs failed. On average, we paid around \$0.12 per job (around \$0.006 per frame). More than 76% of the regular annotations for the tips are within 20 pixels of the groundtruth annotations. Meanwhile, more than 73% of the regular annotations for the apex are within 25 pixels of the groundtruth annotations. We find that annotating the joint and the endpoint of the instrument is a very challenging task. Less than 37% of both annotations are within 25 pixels of the groundtruth annotations.

4.3 Tracking Performance

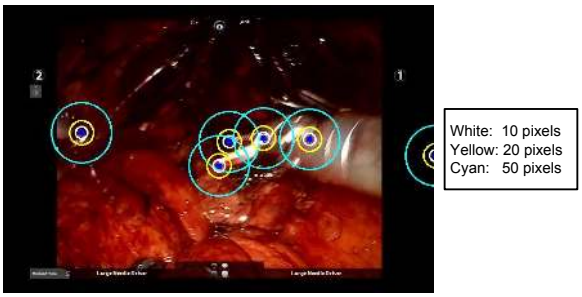

Figure 7: Circles with different radii. The resolution of videos is 720×480. The radii for the white, yellow and cyan circles are 10, 20, and 50 pixels respectively. In our experiments, if the location of a prediction is within 20 pixels of the crowdsourced annotation, we consider it as true positive. Otherwise, we consider it as false positive. This figure is best shown in color.

Tracking the robotic instruments accurately is a crucial part of our computer vision based analysis. Hence, it is important for us to understand the performance of the hourglass network. We evaluate the tracking performance of our network by comparing the predicted locations of the keypoints with the crowdsourced annotations. The network gives a confidence score to each keypoint. If a keypoint of the robotic instruments is not visible in the frame, the network should give a low confidence score to that prediction. The predicted locations of the keypoints should be close to the crowdsourced annotation. Hence, we can treat each keypoint prediction as a binary classification problem and apply the receiver operating characteristic (ROC) curve to evaluate the tracking performance. If the predicted location of a keypoint is within a radius r of the corresponding crowdsourced annotation, we consider it as a true positive. Otherwise, we consider it as a false positive. We show circles with different radii in Fig. 7. In our experiments, we pick r to be 20 pixels.

The area-under-curves (AUC) of the ROC curves for each keypoint are shown in Tab. 2. On average, we can achieve an AUC larger than 0.70 for most of the keypoints of the right instruments. For the left apex, we can achieve 0.82 on average. This shows the hourglass network can achieve a very good performance on tracking the keypoints of the robotic instruments.

	A	B	C	D	E	F	G	H	I	J	K	L	Average
Right Tip 1	0.83	0.87	0.68	0.75	0.65	0.71	0.74	0.72	0.78	0.72	0.64	0.7	0.73
Right Tip 2	0.85	0.88	0.70	0.77	0.68	0.69	0.76	0.73	0.79	0.74	0.66	0.73	0.74
Right Apex	0.79	0.78	0.63	0.69	0.62	0.67	0.73	0.68	0.79	0.68	0.63	0.71	0.70
Right Joint	0.73	0.74	0.68	0.71	0.65	0.68	0.75	0.64	0.76	0.69	0.66	0.73	0.70
Right Endpoint	0.70	0.72	0.64	0.73	0.67	0.62	0.72	0.67	0.73	0.66	0.65	0.70	0.68
Left Apex	0.76	0.87	0.83	0.79	0.87	0.83	0.80	0.80	0.87	0.90	0.73	0.87	0.82

Table 2: AUC of ROC curves for each keypoint.

4.4 Classification Performance

Given the tracking results, we apply SVM to classify the technical level of a surgeon as mentioned in Sec. 3.3. We train and evaluate our SVM using different number of keypoints to see how the number of keypoints affects the performance of classification. We perform the experiments under two settings. In the first setting, we use keypoints annotated by crowd worker. In the second setting, we use keypoints tracked by the hourglass network. These two settings would help us understand how the performance of the tracking task relates to the performance of the classification task. In all our experiments related to SVM, we are using the SVM package by Fan et al. (2008).

	A	B	C	D	E	F	G	H	I	J	K	L	Accuracy
Peer Review (Groundtruth)	lower	higher	higher	higher	lower	lower	higher	higher	higher	higher	higher	higher	-
Human (Right)	higher	higher	higher	higher	lower	lower	higher	higher	higher	lower	higher	higher	83.33%
Human (Right + Left)	lower	higher	higher	higher	lower	lower	higher	higher	higher	higher	higher	higher	100%
HG (Right)	lower	higher	higher	higher	lower	higher	higher	higher	higher	lower	higher	higher	83.33%
HG (Right + Left)	lower	higher	higher	higher	lower	higher	higher	higher	higher	higher	higher	higher	91.67%

Table 3: Performance of the SVM classifier. We train and evaluate the performance of the SVM classifier using the keypoints annotated by human and the keypoints tracked by the hourglass network.

We first perform experiments using the keypoints annotated by crowd workers. As shown in Tab. 3, if we use only the right hand annotations to train our SVM, the accuracy of our SVM classifier in categorizing high or low skill, achieves 83.33% (10 out of 12 videos). If we consider the left apex as well, the accuracy of our SVM classifier in categorizing skill can achieve 100%. This suggests that the movement of the left robotic instrument provides useful information when we classify the skill of a surgeon. We then performed the same experiments using the keypoints tracked by the hourglass network. As shown in Tab. 3, the accuracy for using right instrument is 83.33%, while the accuracy for using both right and left instrument is 91.67%. The performance of the linear classifier on the keypoints tracked by the network is not as accurate when using the keypoints annotated by crowd workers. The quality of the keypoints prediction is crucial to the performance of the classification of technical skill level of a surgeon.

When we train our SVM, we use a sliding window of size 200 and we randomly sample 25000 positive samples and 25000 negative samples for training. We use cross validation in the SVM package to find the optimal parameters for the SVM classifiers. When we test our SVM, we also use a sliding window of size 200 and applied SVM to 2000 windows which are sampled randomly. We then averaged the scores over 2000 windows and make a final prediction for a video.

5. Discussion and Related Work

To the best of our knowledge, although there are studies (Sznitman et al., 2012, 2014), which also propose the use of computer vision algorithms to detect instruments in minimally invasive surgery for 2D detection, our work is the first work proposing the use of crowdsourcing and deep neural networks to not only track instruments but also assess the technical skill level of surgeons performing invasive minimally surgery. Sznitman et al. (2012) proposes a dataset for tracking tool tip in vitreoretinal surgery. The dataset consists of only total 1500 images from 4 sequences of vitreoretinal surgery, meanwhile our dataset contains 146309 images from 12 different surgeons, which is near two order of magnitude larger. Sznitman et al. (2014) proposes a part-based ensemble classifier for detecting different parts of instrument and uses RANSAC to estimate the pose of instrument in minimally invasive surgery. Meanwhile, our networks can be trained end-to-end for both part detection and pose estimation tasks simultaneously. In this work, we not only apply ConvNet for tracking task, but also demonstrate how to apply the tracking results for a higher-level task, the technical skill evaluation task, which has crucial implications for surgical training and evaluation.

The value of studying video data is that it is applicable to all minimally invasive platforms. The fundamental nature of the work we have undertaken has the potential to be applied to endoscopic, laparoscopic and robotic surgery videos, which are widely available. Just as prior work has shown that laypeople can identify a highly skilled surgeon from a lower skilled surgeon (Ghani et al., 2016), we predict that in the future, computer vision methods will be able to do the same thing. If implemented broadly in surgical disciplines, computer vision analysis of technical skill would have significant implications for multiple stakeholders. A project of this significance has the potential to provide a scalable, rapid and objective method for the assessment of both laparoscopic and robotic surgery, and our findings could have an important impact for surgical education, credentialing, hiring and quality improvement. As Ghani et al. (2016) points out, "Better skills may lead to improved patient care, which would ultimately benefit physicians, patients, and payers".

There are other studies (Allan et al., 2013, 2015) proposing methods for 3D instrument pose estimation. The proposed method in this work is currently limited to 2D trajectories, without reconstructing the 3D trajectories. This limits our algorithm to assess the technical skill of a surgeon related to depth, such as "Depth Perception" and "Force Sensitivity", when we evaluate the technical skill level of a surgeon.

Moreover, our method currently focuses on tracking the instruments and classifying the surgeons into two skill level groups by extracting different features from the movement of robotic instruments. There is more to understanding surgical skill than the ability to track surgical instruments. There are other features we have not utilized in this work such as the color (i.e. may indicate whether there is excessive bleeding). Different kind of features other than instruments movement may help improve the performance of the classification. To better evaluate the surgical skill of the surgeons, in our future work, we would also like to develop computer skill score metric similar to GEARS and extend our work to assess relationship between computer vision derived skill, with peer surgical skill score and patient outcomes.

Acknowledgments

This work is supported by Intuitive Surgical under a Technology Research Grant. We would also like to acknowledge the significant contribution of Blue Cross Blue Shield of Michigan, which supports Michigan Urological Surgery Improvement Collaborative, and thank all robotic surgeons who have contributed videos to this program.

References

Max Allan, Sébastien Ourselin, Steve Thompson, David J Hawkes, John Kelly, and Danail Stoyanov. Toward detection and localization of instruments in minimally invasive surgery. *IEEE Transactions on Biomedical Engineering*, 60(4):1050–1058, 2013.

Max Allan, Ping-Lin Chang, Sébastien Ourselin, David J Hawkes, Ashwin Sridhar, John Kelly, and Danail Stoyanov. Image based surgical instrument pose estimation with multi-class labelling and optical flow. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 331–338. Springer, 2015.

Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693, 2014.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9 (Aug):1871–1874, 2008.

Khurshid R Ghani, David C Miller, Susan Linsell, Andrew Brachulis, Brian Lane, Richard Sarle, Deepansh Dalela, Mani Menon, Bryan Comstock, Thomas S Lendvay, et al. Measuring to improve: peer and crowd-sourced assessments of technical skill with robot-assisted radical prostatectomy. *European urology*, 69(4):547–550, 2016.

Alvin C Goh, David W Goldfarb, James C Sander, Brian J Miles, and Brian J Dunkin. Global evaluative assessment of robotic skills: validation of a clinical assessment tool to measure robotic surgical skills. *The Journal of urology*, 187(1):247–252, 2012.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Jeffrey J Leow, Steven L Chang, Christian P Meyer, Ye Wang, Julian Hanske, Jesse D Sammon, Alexander P Cole, Mark A Preston, Prokar Dasgupta, Mani Menon, et al. Robot-assisted versus open radical prostatectomy: a contemporary analysis of an all-payer discharge database. *European urology*, 70(5):837–845, 2016.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

Alejandro Newell and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. *arXiv preprint arXiv:1611.05424*, 2016.

Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

Ben Sapp and Ben Taskar. Modec: Multimodal decomposable models for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3681, 2013.

Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014a.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014b.

Raphael Sznitman, Karim Ali, Rogério Richa, Russell Taylor, Gregory Hager, and Pascal Fua. Data-driven visual tracking in retinal microsurgery. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*, pages 568–575, 2012.

Raphael Sznitman, Carlos Becker, and Pascal Fua. Fast part-based classification for instrument detection in minimally invasive surgery. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 692–699. Springer, 2014.

Carl Vondrick, Donald Patterson, and Deva Ramanan. Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision*, 101(1):184–204, 2013.