International Journal of Computer Assisted Radiology and Surgery (2022) 17:437–448
https://doi.org/10.1007/s11548-022-02559-6

ORIGINAL ARTICLE

Video-based fully automatic assessment of open surgery suturing skills

Adam Goldbraikh[1] · Anne-Lise D'Angelo[2] · Carla M. Pugh[3] · Shlomi Laufer[4]

Received: 26 April 2021 / Accepted: 3 January 2022 / Published online: 1 February 2022

Abstract

Purpose The goal of this study was to develop a new reliable open surgery suturing simulation system for training medical students in situations where resources are limited or in the domestic setup. Namely, we developed an algorithm for tools and hands localization as well as identifying the interactions between them based on simple webcam video data, calculating motion metrics for assessment of surgical skill.

Methods Twenty-five participants performed multiple suturing tasks using our simulator. The YOLO network was modified to a multi-task network for the purpose of tool localization and tool–hand interaction detection. This was accomplished by splitting the YOLO detection heads so that they supported both tasks with minimal addition to computer run-time. Furthermore, based on the outcome of the system, motion metrics were calculated. These metrics included traditional metrics such as time and path length as well as new metrics assessing the technique participants use for holding the tools.

Results The dual-task network performance was similar to that of two networks, while computational load was only slightly bigger than one network. In addition, the motion metrics showed significant differences between experts and novices.

Conclusion While video capture is an essential part of minimal invasive surgery, it is not an integral component of open surgery. Thus, new algorithms, focusing on the unique challenges open surgery videos present, are required. In this study, a dual-task network was developed to solve both a localization task and a hand–tool interaction task. The dual network may be easily expanded to a multi-task network, which may be useful for images with multiple layers and for evaluating the interaction between these different layers.

Keywords Surgical video data · Tool localization · Surgical Simulation · Motion metrics

Introduction

Since Dr. Erich Mühe performed the first laparoscopic cholecystectomy in 1985, it has become the gold standard surgical treatment for gallbladder disease [48]. Nevertheless, conversion to open surgery may be necessary in cases of medical and surgical complications related to anesthesia, peritoneal access, pneumoperitoneum, and thermocoagulation [21]. Cholecystectomy is just one of many examples of open surgery procedures that are being replaced by minimal invasive surgery (MIS), yet may require reverting to open surgery in the face of complications. Thus, while the new generation of surgeons has less experience with open surgery procedures [11,38], they still must master open surgery skills to handle the more extreme situations [14].

The recent advances in deep learning and computer vision have led to a growing number of studies focusing on automatic analysis of surgical video data [1,56]. Since the use of video is an integral part of MIS, most of these studies have focused on laparoscopic and robotic surgery. In contrast, video capture is not well established in open surgery [51]. Thus, open surgery has not benefited from the many advantages computer vision and deep learning methods have

✉ Adam Goldbraikh
sgoadam@campus.technion.ac.il

Anne-Lise D'Angelo
annelise.dangelo@gmail.com

Carla M. Pugh
cpugh@stanford.edu

Shlomi Laufer
laufer@technion.ac.il

1 Applied Mathematics Department at the Technion, Israel Institute of Technology, 3200003 Haifa, Israel

2 Mayo Clinic, Surgery, Rochester, MN, USA

3 School of Medicine, Stanford University, Stanford, CA, USA

4 Faculty of Industrial Engineering and Management at the Technion, Israel Institute of Technology, 3200003 Haifa, Israel

to offer for skill training and automatic assistance. Furthermore, while the use of simulators is inherent to training MIS [16], assessing open surgery skills, which is no less important, is lagging [9,11]. This study will focus on both the development of novel video analysis algorithms as well as using these algorithms for assessment of surgical skills.

There are some fundamental differences between the video data obtained during MIS and during open surgery. In MIS the video image usually includes 1–2 tool tips which are in actual use. Therefore, *tool presence detection* and *tool localization* are common goals in multiple studies analyzing MIS [27,56]. Tool presence detection refers to identifying the existence of the tool in the image, while tool localization indicates providing its position as well. In contrast, the video image during open surgery will often show not only the tool tip but also the hand. The image may include 2–4 hands captured concurrently in a range of positions and activities as well as stationary tools not held by anyone. For example, one surgeon might have a needle driver loaded with a needle in one hand and forceps lifting the tissue in the other hand. Meanwhile, another surgeon may be assisting by stabilizing the tissue with one hand and holding scissors in the other. Generating just a list of all the objects present in the image ignores the interaction between the objects and thus provides only a partial description of the image. Full analysis of the image structure should include the identification of the different tools and hands as well as their interactions.

The traditional teaching and assessment of technical skill relies heavily on the apprentice model, in which residents perform the procedure on a patient in the operation room (OR) under the guidance and evaluation of an expert. This approach does not provide a standardized method for training and assessing surgical skills [8]. This led to the development of simulation-based training and assessment. Traditionally, this included observer-generated task-specific checklists and global rating scales. Both methods are time-consuming and tend to bias [6]. The crucial need for objective methods has motivated the development of technology-based approaches [6,39,46,49].

In recent years, there is a growing interest in methods for tele-education and tele-simulation [41,50]. Furthermore, with the recent outbreak of the Coronavirus (COVID-19) pandemic, the need for novel methods of remote education in general and the training of surgeons in particular has become clearer than ever [20,37,53]. Sensor-enabled simulations may be integrated with remote education, thus providing objective assessment and feedback. Yet, they typically require expensive equipment and a complex setup, which is more appropriate for modern simulation centers and not for the home environment. This need to develop reliable surgical simulations that use cheap technology is not a new concern; it has been coming up in the context of developing countries where the resources are limited [3,23,31,58]. Therefore, in this study we will evaluate a system that enables self-training and assessment of open surgery technical skills at the home of the trainee. With such a setup in mind, we captured video data using a standard webcam connected to a laptop. The algorithms developed are fast and can be analyzed on the cloud or even locally on the CPU or GPU within a reasonable processing time, providing evaluation scores in a timely manner. The simulator used includes a simulation board and basic surgical tools which can be supplied by mail.

The technical goal of this work is twofold: first, study both surgical tool localization and surgical image structure; second, evaluate video-based kinematic analysis of technical surgical skills. The contributions of our work are the following. We developed a variable tissue simulator which we use for assessment of open surgery skill [6,7]. Task analysis of the video revealed that open surgery video data require new categories which were not defined in previous studies on MIS video data. The task analysis was followed by the development of a new near-real time multi-task detection network for detecting the position of all the tools and hands in the image as well as identifying which tool is being used by each hand. We used the output of these algorithms to assess surgical performance based on multiple motion metrics. Finally, our multi-task system provides hand location, tool location, and hand–tool interactions. This combined knowledge led to the development of a new motion metric that examines the technique used for holding the tool.

Related works

Multiple studies have demonstrated that kinematic data can provide valuable information in the assessment of surgical skill [39,46,49]. However, in open surgery, most studies use kinematic data from sensors such as electromagnetic 6DOF sensors for the measurement of hand motion [7,39,46,49]. These sensors are typically expensive and may have a complex setup as well as interfere with the normal workflow. Based on the kinematic data, different skill-evaluation metrics are calculated, such as procedure time, path length, number of hand movements, working volume. [6,7]. Each metric can indicate different aspects of motor skills level; in this sense, these types of models are highly explainable.

Simulators measuring kinematic data have been developed for robotic and minimally invasive surgery as well [46]. Several public datasets contain robotic kinematic data with skills assessment labeling such as JIGSAWS and MISTIC-SL [18,19]. Machine learning methods that predict the surgeon's level of expertise based on these metrics have been developed [12,61], as well as deep learning methods that predict the level of expertise directly on the data (kinematic or video) without intermediate feature calculation [13,15]. Kinematic data can be obtained also by using computer vision meth-

ods upon video data, e.g., by using bounding boxes of object detection [28] or landmark detection [10] on hands or tool tips.

Analysis of MIS video data has raised a range of research questions. Evaluation of topics such as tool presence detection [28,30], workflow recognition [5,42], error identification and skill assessment [28,40] has the potential of making the surgical environment safer and more efficient. In a recent study, detection of hands in open surgery videos was assessed [60].

The goal of this study is to detect tools and hands. Therefore, the selection of the optimal object detection algorithm, which is the engine of our system, is of the utmost importance. Current object detection algorithms are generally grouped into two main families: two-stage algorithms and one-stage algorithms. In the two-stage algorithms, the first stage extracts regions of interest, namely those regions where the objects are expected to be found. The second stage involves classifying the objects and locating their bounding boxes. Two stage algorithms, such as Faster R-CNN [47] and Mask R-CNN [25], are characterized by high accuracy rates and long run time. As a result, they are not appropriate for real-time applications. In contrast, the one-stage object detection algorithms do not require intermediate steps, as they frame the object detection as a single regression for both identifying bounding boxes and classifying them in one stage. These algorithms are less accurate than the two-stage algorithms, but because they work much faster than the two-stage algorithms, they are more suitable for real-time applications. This family includes the SSD algorithm [35], RetinaNet [34] and all versions of YOLO [43–45]. A new sub-family of one stage object detection algorithms has been recently introduced. These algorithms are based on the transformer architecture and involve a single stage. While this renders them more accurate, they are slower and therefore not suitable for real-time applications [4,36]. In general, if fast and accurate detection is required, the one-stage YOLOv3 is considered as a great choice. For example, YOLOv3 was used for real-time Jellyfish classification [17], real-time people detection [24], and real-time pattern-recognition of ground-penetrating radar images [32]. In the surgical tool detection domain, two-stage object detection algorithms have been implemented, such as R-CNN based networks [28] as well as one-stage algorithms where inference time is significant, such as YOLO9000 in [29] and RetinaNet in [60] and SSD in [2].

There are some topics in the computer-vision community that can be considered as related to the unique challenges of open surgery, such as hand–object interaction [52], human–object interaction [33] and object–object interaction [26]. In [26], the authors propose inter-object graph representation for recognition of activities in self-driving scenarios. Their method is based on disentangled graph embedding with direct edge appearance observation. Their most relevant observation to our work is that relations between objects are captured in a single bounding box that contains both interacted objects strongly rather than in using tight boxes of the objects separately.

Methods

The first goal of this work was to detect the position of all the tools and hands in the image; that is, provide *tool localization*. The training set contained images with labeled objects and their locations. We specified the object's location by bounding it with a tight box. In the labeled images, all the tools and all the hands (whether holding a tool or not) were outlined. We used a YOLO detection network [45] for the *tool localization* task. It should be noted that in some cases the hand is empty, for example when palpating tissue. In this case, the hand may be regarded as the "tool." However, in most cases the hand is holding a tool. For simplicity, when we use the term *tool localization*, we mean all the tools and all the hands in the image.

The second goal was to determine the *hand–tool interaction*. In a previous study [22], we used the output of the *tool localization* algorithm combined with spatial assumptions to match between the hand and the tool. For each hand detected we examined whether there was a tool in close proximity and if so, it was assumed it was being used by that hand. The labeling for such a task was simpler. We only needed to annotate start and end time of each tool usage. Therefore, for this task we labeled the full dataset (whereas for *tool localization*, we only labeled a subset of the images, as in other deep learning studies).

However, this approach is based on heuristic assumptions and prone to manual fine tuning. In this study, we will develop a standard approach using deep learning to provide a general solution. For this, we added another layer of annotation. In each image annotated for the *tool localization* task we added another set of bounding boxes. In this new set, each pair of hand+tool is outlined using a tight box (Fig. 1). Now we can use a detection network such as YOLO for determining the *hand–tool interaction* as well. It should be noted that now we have two sets of ground truth labels for the *hand–tool interaction* task. The first includes the entire video set; however, it includes only the interaction and not its spatial position. The second set includes only a small sample of the images. This set includes bounding boxes and is used for training and testing using traditional machine learning approaches. There are two naïve methods to train the network to provide the complete image structure (*tool localization* and *hand–tool interaction*). The first is to train one network for detecting the tools and another, separate network for detecting the combination of *hand–tool interaction*. As we will show in

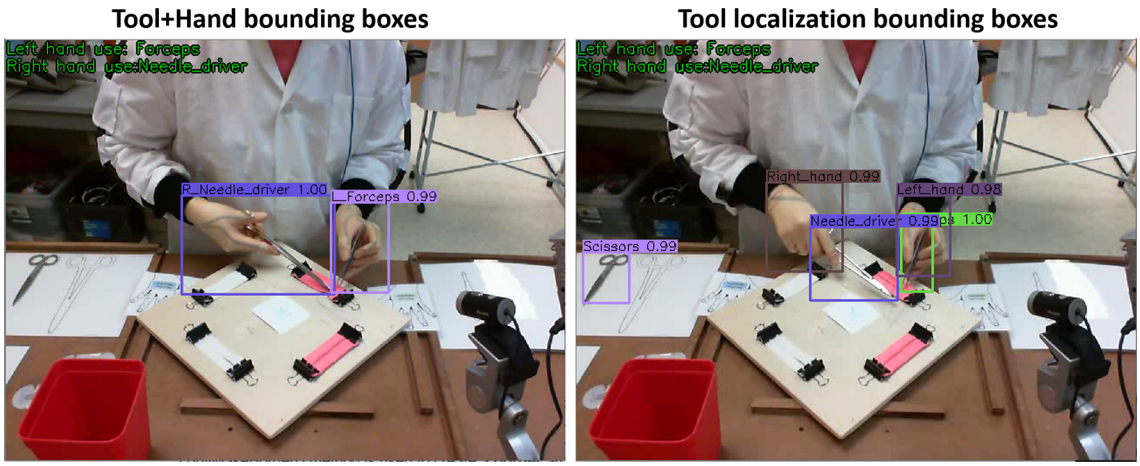

Fig. 1 Each bounding box in the left figure represents a hand–tool interaction object, and every box in the right figure represents tool or hand. The text in the top-left corner outputs the final decision on the tool usage

the results, this method provides a good outcome; however, analysis requires twice the computation power. The second approach would be to combine all the labeled data (the tools, the hands and the hand+tool) then train one network with the entire dataset. This approach saves computation power; however, as we will see, it leads to a significant decrease in accuracy.

Therefore, in this study we developed a multi-task detection network to solve both (*tool localization* and *hand–tool interaction*) challenges. The multi-task network is based on the YOLO network; however, we updated the final layers to support multiple detection tasks. Using this new network, we gain from both worlds: our detection is as good as two separate networks while the required computation power is only slightly more than one network.

Variable tissue simulator

The variable tissue simulator was developed to simulate a suturing task to assess decision making during suturing tasks of varying difficulty. The simulator consists of a board to which simulated material is connected by two clips [6]. The task was to place three interrupted instrument-tied sutures on two opposing pieces of the material. Two different materials were used: tissue paper simulating friable tissue, and rubber balloons simulating arteries. Each participant was provided with three tools: a needle driver, surgical forceps, and suture scissors. Top view video data were captured in a frame rate of 30 FPS.

Data collection

Eleven medical students, one resident, and 13 attending surgeons participated in the study. Each participant performed twice on the friable tissue simulator and twice on the artery simulator. Thus, there were a total of 100 videos, each approximately 2–6 min long. This dataset was split into two sets. Where the first contains 15 videos, we will refer to it as *train video set* and the second, *test video set*, contains the rest. From the train video set 924 frames were picked and split to seven sub-sets of 132 images for k-fold cross-validation. Each selected frame was labeled with two sets of bounding boxes: the first for the *tool localization* task and the second for the *hand–tool interaction* task. In addition, from 5 other videos, 200 frames were chosen for a *test set*. These frames were labeled in the same method. The labeling was performed with Microsoft's Visual Object Tagging Tool. Finally, the entire video dataset was labeled for the start and end time of each tool usage using Behavioral Observation Research Interactive Software (BORIS). As mentioned, we have two sets of ground truth labeled data. The first includes tight bounding boxes and is used for the training and testing of the different classifiers. In this set, we require an IoU of at least 0.5 with the ground truth bounding box to be considered as a true prediction, and average precision (AP) is used to assess the results. The second labeled dataset includes all the data; however, it does not include any bounding boxes, only start and end point. Therefore, this set is only used for testing the *hand–tool interaction* performance. This is assessed using precision, recall, and F1 metrics.

The following data augmentation was used during the training. Horizontal flip applied with a probability of 0.5. We uniformly randomly rotated the images and automatically fit corresponding bounding boxes, based on its transformation matrix in the range of $\pm 7°$. In addition, we use the standard PyTorch Torchvision ColorJitter module, which randomly changes the color values with the following parameters: brightness = 0.2, contrast = 0.2, saturation = 0.1, and hue = 0.05. The training and evaluation were performed on a

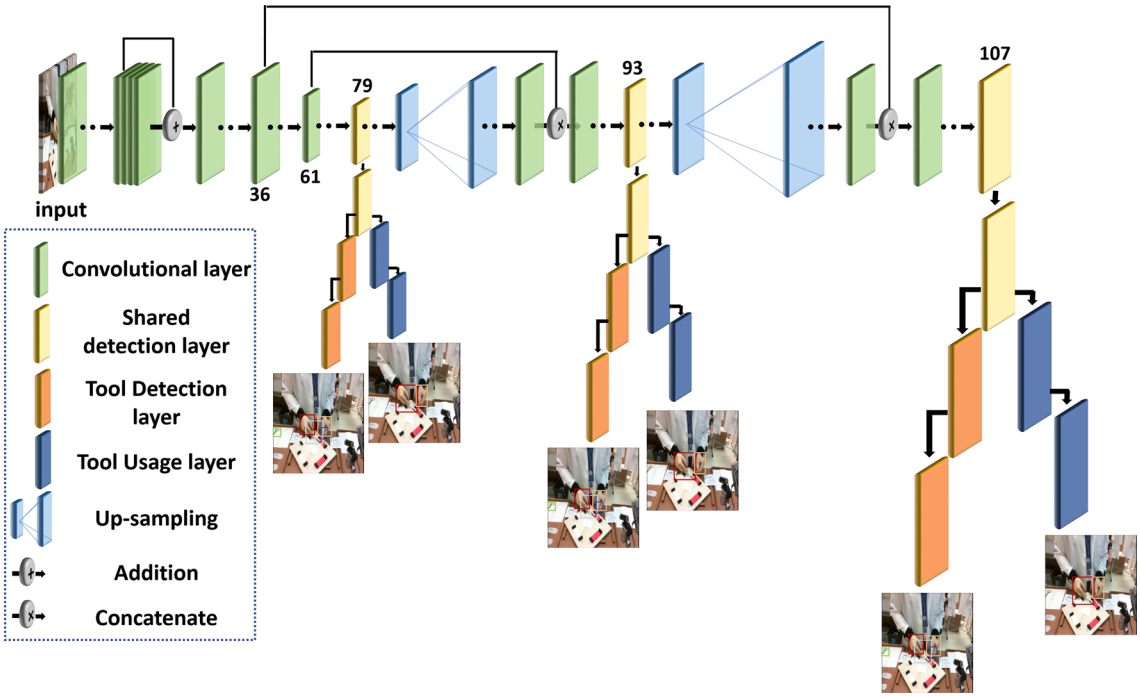

Fig. 2 Schematic illustration of proposed architecture

NVIDIA Tesla V100 Volta GPU Accelerator 32GB Graphics Card.

Naive approaches

Two naive approaches may be used for solving the *tool localization* and *hand–tool interaction* problems. The first approach is to train two separate networks, one for each problem. One network will include five categories of tool types—D and the second the eight classes of *hand–tool interaction* combinations—S. The second approach is to train one integrated system for both problems. In this case, the merged fourteen categories—$D \cup S$ will be used. For all tasks, a YOLOv3 (YOLO) network was used.

All three systems were trained under similar conditions: Adam optimizer up to 400 epochs with learning rate of 10^{-3} and additional 200 epochs with learning rate of 10^{-4}. The model selected was the model with the highest AP with respect to the validation set. During the test session, each frame was tested twice, the first time with a horizontal flip and the second time without. The models are tested on the 200 test frames as defined in the previous section.

Multi-task deep neural network approach

Our architecture is an extension of the YOLO network [45]. YOLO is a fully convolutional network that consists of 106 layers. To provide detection at multiple image scales YOLO uses multiple detection heads. Each head consists of four convolutional layers and one YOLO prediction layer. The input image size is 416×416, and three color channels are used. In our new architecture, each detection head was split after the second convolutional layer. We will refer to the layers after the split as branches. Hence, every original detection head was split into two branches: a *hand–tool interaction* branch and a *tool localization* branch (see Fig. 3). The output of each branch is shown in Fig. 1. We will refer to all layers that are accessible to both branches as the trunk of the network. The new architecture is depicted in Fig. 2. YOLO uses *non-maximum suppression* (NMS) to address the problem of multiple detections of the same object. For the *tool localization* branch, an additional NMS was added. It suppressed multiple tools in the same area while allowing for tools and hands to overlap.

Training method

Technically, we have two datasets: one for the *tool localization* task and the other for the *hand–tool interaction* task. Note that the two datasets are based on the same set of images and differ only by the tagging. Each epoch contains batches from the two datasets. The batches are ordered in round-robin fashion, where after each *tool localization* batch was placed, a *hand–tool interaction* batch was used. While training for one task, the branch dedicated to the other task was frozen. The network was trained by using Adam optimizer with a learning rate of 10^{-3} for the first 200 epochs and a learning rate of 10^{-4} for an additional 100 iterations. We trained in

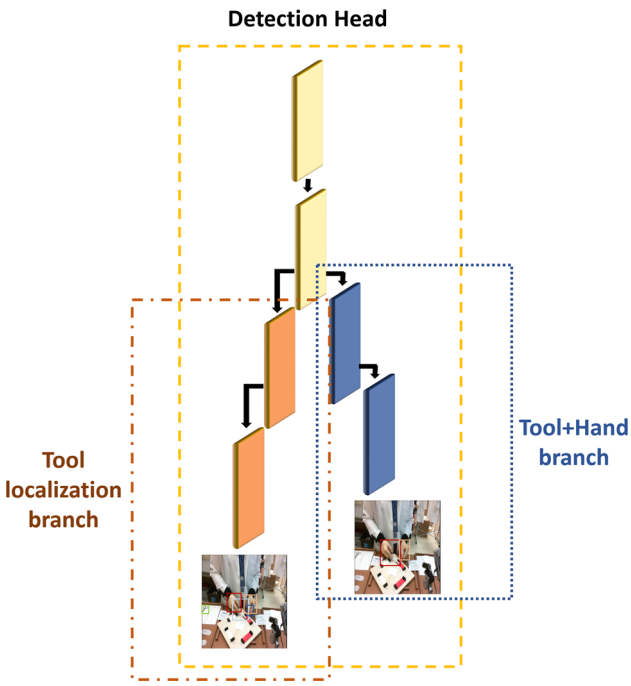

Fig. 3 The split in the detection head to *tool localization* branch and *hand–tool interaction* branch

k-fold cross-validation manner ($K = 7$), where every model, in addition to its validation, also tested on our test-set.

Inference of tool usage

The goal of the *hand–tool interaction* was to provide exactly one bounding box for each hand visible in the image. When unsuccessful, data from the *tool localization branch* may be used to help infer tool usage. This includes two scenarios (see Fig. 4):

Scenario 1—no bounding box

This scenario includes situations in which the *hand–tool interaction* branch provides no bounding box for one of the hands, yet the *tool localization* branch detects that hand. In this case, we search the output of the *tool localization* branch for an overlap between the hand's bounding box and one of the tool's bounding boxes, described more specifically in [22].

Scenario 2—multiple bounding boxes

This scenario includes situations in which the *hand–tool interaction* branch provides multiple bounding boxes of one of the hands, and we need to select the correct bounding box. All the bounding boxes for that hand will be compared with all the bounding boxes of all the tools detected by the *tool localization* branch; the pair with the largest overlap will be selected.

Smoothing process

For each frame, the final *tool–hand interaction* was based on the majority of the previous 15 frames. In addition, empirical data revealed that when the hands were moving fast, *tool localization* was significantly reduced due to image blurriness while hand detection was not reduced. Therefore, hand speed was calculated based on hand detection data. Decisions regarding *tool–hand interaction* were not changed during fast movement. The smoothing process only influenced the decision of which tools were being used at any given moment and did not affect the bounding boxes provided by the system.

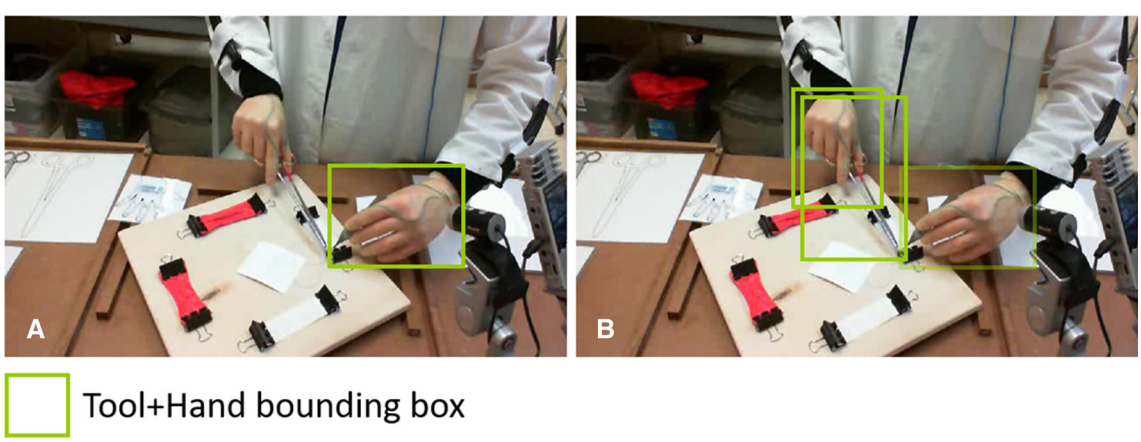

Fig. 4 **a** Scenario 1: No *hand–tool interaction* bounding box for the right hand, **b** Scenario 2: There are two *hand–tool interaction* bounding box for the right hand

Table 1 Average precision (AP_{50}) of each class in the tool localization network, Tool+Hand network, combined results of tool localization network and Tool+Hand network, and integrated network with all 13 classes. The results are with respect to the test set

	Tool localization (AP_{50})	Tool+Hand (AP_{50})	Regular Yolo (13) with all classes (AP_{50})
Right hand	0.956		0.766
Left hand	0.947		0.783
Needle driver	0.839		0.740
Forceps	0.728		0.659
Scissors	0.845		0.843
mAP (Tool localization only)	**0.863**		**0.758**
Scissors in right hand		0.807	0.731
Scissors in left hand		0.827	0.751
Needle driver in right hand		0.927	0.916
Needle driver in left hand		0.907	0.876
Forceps in right hand		0.918	0.854
Forceps in left hand		0.954	0.839
Empty right hand		0.787	0.001
Empty Left hand		0.838	0.000
mAP (Tool + Hand only)		**0.871**	**0.621**
mAP (Both tasks)	**0.868**		**0.674**

Results

Results naive approaches

Table 1 summarizes the results for the naive approaches. The tool localization only network yielded mAP of 0.863 and the integrated network achieved only 0.758 on these classes. The Tool+Hand only network has mAP of 0.871 on it's 8 classes when the combined network has only 0.621. In total, the separate networks show significantly better results, with mAP of 0.868 than the one integrated system, with mAP of 0.674.

Results multi-task deep neural network approach

The performance of the multi-task system in the *tool localization* framework is depicted in Table 2. The detection was slightly better than the two separate networks; this is a known effect of multi-task training. The system yielded an expectation mAP = 0.874 for the *tool localization* task, mAP = 0.885 for the *hand–tool interaction* task, and the overall results were mAP = 0.881. The results are based on the 200 *test set* images. In the results table, one can see also the validation set results.

Tool–hand interaction was also tested using the 85 full videos from which frames were not taken for the training process. For 3.92% of the data the *hand–tool interaction* branch was unsuccessful and tool usage was inferred using the data from the *tool localization* branch. The precision, recall, and F_1 of each tool as well their average values and the total accuracy were calculated (Table 3).

Motion-based metrics for surgical skills assessment

In this section, we use our algorithm to develop a fully automatic skills assessment system. In our prior studies, we have measured performance using motion metrics such as procedure time, path length, number of hand movements, and working volume. [6,7] In these studies, sensors were used to measure the hand position. Here, video data were used to analyze the motion pattern and then derive the metrics. Most sensor systems provide three-dimensional motion data dimensions. Our method relies on two-dimensional projection of the motion data on the image plane. Nevertheless, we get statistically significant separation between medical students and experts.

Metric computation

The three traditional metrics calculated are duration, path length, and number of movements. The output of the *hand–tool interaction* algorithm is used to define the duration of the procedure. We define the beginning of the procedure as the first frame in which at least one hand is using a tool and the end of the procedure as the last frame in which one of the hands is using a tool. The procedure duration is calculated as the total number of frames divided by 30. The location of an object is defined as the center point of its bounding box. Path length is the two-dimensional distance the hands moved, in the image plane, from the starting point until the end of the suturing task.

The velocity v of each hand is calculated as $v = \sqrt{v_x^2 + v_y^2}$, where v_x an v_y are the first order numerical derivatives calculated by the centered difference formula on x and y position vectors. The hand is considered as static when the velocity is below the threshold value of $25 pixel/sec$. Finally, the number of movements is defined as the number of times velocity crosses the threshold value divided by two.

In addition to the three traditional metrics in this study, we define two new metrics. Both metrics assess the holding angle of the forceps. The holding angle of the forceps was defined as the aspect ratio of the bounding box of the forceps $width/height$. The first metric is the mean of the aspect

Table 2 The validation and test results of *Tool localization* and *hand–tool interaction* results (sevenfold cross-validation). All values are in terms of AP_{50}

	Tool+Hand Detection								
	Scissors in right hand	Scissors in left hand	Needle driver in right hand	Needle driver in left hand	Forceps in right hand	Forceps in left hand	Empty right hand	Empty left hand	mAP
Validation (AVG+SD)	0.896± 0.035	0.890± 0.053	0.955± 0.028	0.954± 0.02	0.898± 0.04	0.901± 0.041	0.909± 0.020	0.91± 0.02	**0.914± 0.017**
Test (AVG+SD)	0.849± 0.045	0.820± 0.051	0.926± 0.009	0.917± 0.013	0.915± 0.018	0.92± 0.031	0.87± 0.011	0.864± 0.014	**0.885± 0.012**
	Tool localization								
	Right hand	Left hand	Needle driver	Forceps	Scissors				mAP
Validation (AVG+SD)	0.932± 0.012	0.920± 0.013	0.937± 0.016	0.866± 0.038	0.959± 0.028				**0.923± 0.015**
Test (AVG+SD)	0.937± 0.01	0.933± 0.009	0.856± 0.014	0.756± 0.008	0.890± 0.016				**0.874± 0.004**

Table 3 Tool usage task results

Tool	Not smoothed			Smoothed		
	Precision	Recall	F1	Precision	Recall	F1
Left hand	0.606	0.805	0.692	0.693	0.808	0.746
Right hand	0.942	0.949	0.945	0.943	0.959	0.950
Needle driver	0.970	0.946	0.958	0.972	0.957	0.965
Forceps	0.893	0.906	0.900	0.909	0.907	0.908
Scissors	0.842	0.683	0.754	0.874	0.755	0.810
Mean	**0.850**	**0.858**	**0.850**	**0.878**	**0.877**	**0.876**
Accuracy		**0.924**			**0.935**	

ratio throughout the procedure. The second is the standard deviation of the aspect ratio.

Results

As anticipated from previous studies, the attending surgeons performed the task in less time, shorter path length, and smaller number of movements (Fig. 5). The attending surgeons held the forceps at approximately 45° while the students held it in a more upright position. Furthermore, lower standard deviation was measured for attending surgeons, suggesting a more stable and consistent grip of the forceps. Since this is a new metric, further analysis is required for providing accurate interpretation of these findings. However, the method might provide new valuable information for the training of new surgeons.

Discussion

This study focused on the analysis of video data captured using a webcam during open surgery simulation. Our premise was that in open surgery, in addition to detecting the presence of the tools and hands, their interaction needs to be identified. We examined two naive approaches, performing both tasks using one YOLO and using a full network for each task. Since both approaches aimed at solving a standard object detection problem, we would have expected that the results of the integrated system be close to the combined results of the two separate systems. However, the two separate networks yielded significantly better results. A possible explanation for the decreased performance of the integrated network in comparison with the separated networks may be the extended feature overlap between corresponding classes. While using two different networks to solve the two tasks provided good results, this solution is expensive in terms of run time, which is critical for fast simulation analysis.

Therefore, a dual-task network was constructed and a training scheme was developed. The dual-task network performance was similar to that of two networks, while computational load was only slightly bigger than one network. The system was capable of analyzing approximately 35 frames per second on a NVIDIA Tesla V100 Volta GPU and 15 frames per second on a NVIDIA GeForce GTX 1060 GPU. Thus, feedback may be provided in a timely manner.

We chose YOLO as our base network due to its short runtime. The network performed very well on our dataset and achieved a mean average precision (mAP) of 87.4 for tool detection. Jin et al. [28] trained a slower two-stage object detection network based on faster R-CNN to detect laparoscopic tools and reached a mAP of 63.1. The data analyzed in that study were more complex than our data. While our data were captured with a static camera in a simulation environment, in [28] the data were the m2cai16-tool-locations dataset, which includes videos taken during cholecystectomy by a mobile endoscopic camera. This suggests that net-

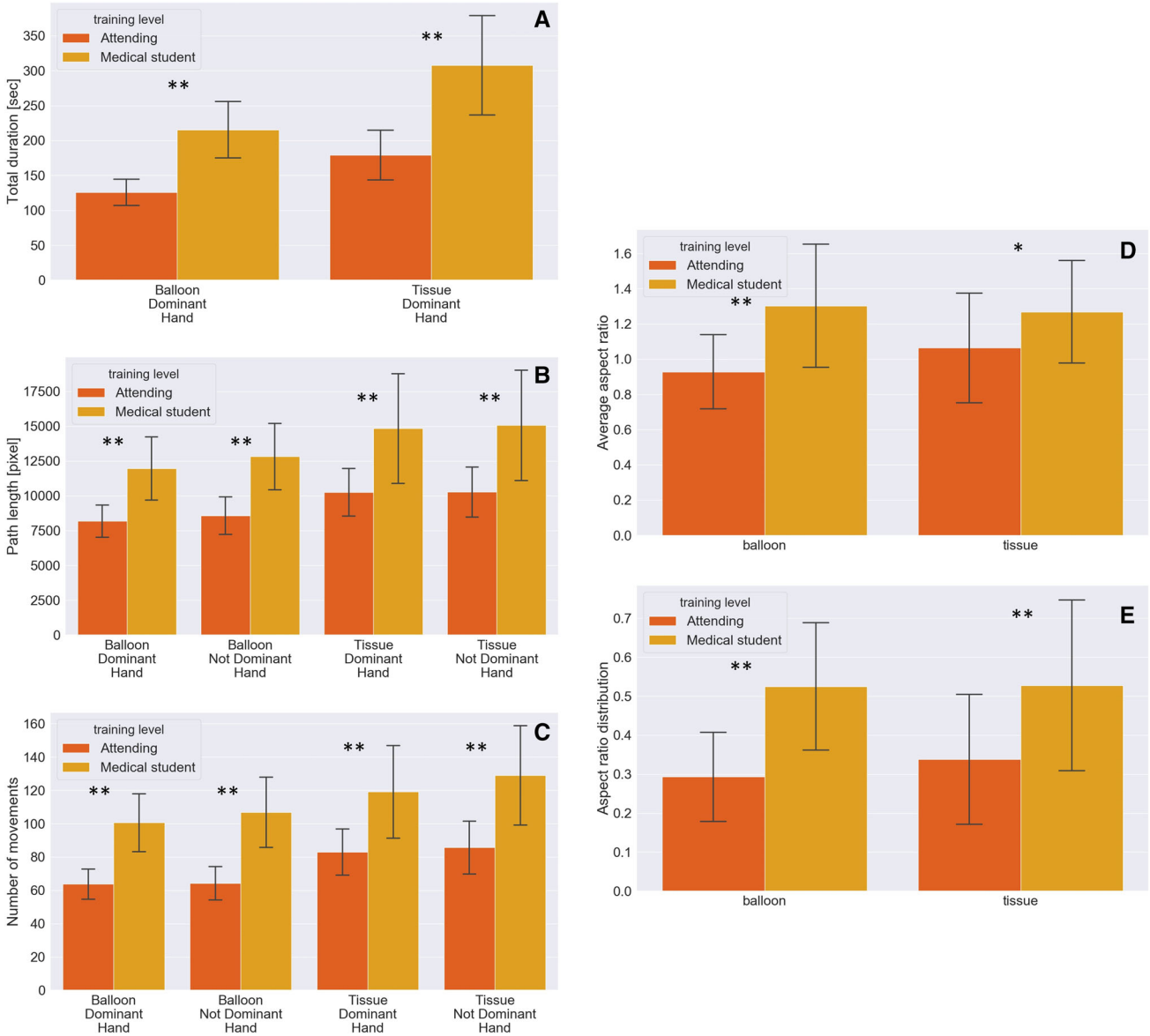

Fig. 5 **a** Procedure duration, **b** path length, **c** number of movements, **d** average mean of the aspect ratio of the forceps' bounding box during its usage, **e** average standard deviation of the forceps' bounding box during its usage. $*p$-value < 0.05 and $**$ p-value < 0.01

work selection may be based on the combination of run-time limitations and data complexity. This option was explored by Soviany et al. [54]. In this study, an image difficulty predictor was developed. Based on the assumed difficulty the system decided whether two-stage object detection was required for an image or whether good performance could be achieved with a one stage detector. This approach may balance between run-time limitations and detection requirements.

The dual-task approach developed in this study may be easily expanded to a multi-task approach if needed. A multi-task system could identify multiple structures in an image. For example, in the surgical context, it could help identify the arm-hand interaction and tool–hand interaction as well as identify if the forceps are holding a needle. This approach is not limited to the surgical arena. In [59], basketball movements and pass relationships were studied. For the task, two separate YOLO networks were trained, one for the players (with and without the ball) and one for their jersey number. This could have been done using one dual-task network, saving run-time. Another example comes from the field of autonomous driving, where analysis of critical events often depends on object–object interaction between cars, pedestrians, road signs, and other prominent objects, and naturally must be evaluated in real time [26].

Video-based motion analysis is a much cheaper and simpler approach than sensor kinematics based assessment. Sensor systems (such as 3D Guidance 6DOF Sensors) may cost thousands of US dollars, while in this study a simple webcam was used. In addition, connecting the participants to the sensor system is time consuming. Using RGBD cameras and LIDAR technology may provide a good combination of 3D information, fast setup time, and low prices. In previous works, these technologies were used for action recognition in OR, medical training, and assessment [55,57]. However, these technologies are not available at most households. Webcams are nowadays a standard component of every computer station or laptop. In addition, due to the COVID-19 outbreak people feel very comfortable in operating this technology. Using a webcam, measuring performance may be as easy as recording a Zoom session. Therefore, the system provides the opportunity for fully automated skill assessment that may be used by the resident or medical student independently.

Although the system only captured two-dimensional data, our data showed significant differences between experts and novices. Furthermore, in addition to the traditional motion metrics, we identified new metrics that might suggest different techniques for holding the tools and perhaps provide more detailed feedback for improvement. Tool orientation showed that experienced surgeons hold the tool differently than medical students. Moreover, the variability over time of tool orientation was significantly higher for medical students. This suggests they are still exploring the optimal holding technique while experienced surgeons have developed a stable approach. Nevertheless, this is a new metric and more work is required to fully understand it.

One limitation of our study was that we used the same webcam. For the system to be generalized for domestic use, the DNN should be trained using data from a wide range of cameras. The focus of this current study was to demonstrate that a webcam may be used for assessing technical skill. Thus, data from 13 attending surgeons were captured. This limited our work to a hospital area, and since the focus was comparing experts and novices, we kept the system standard. The next phase of our work will be to collect data using multiple systems. This may be done by the medical students using their own equipment.

In this study, we focused on data collected from a medical simulator. While this provides independent merit as an approach for assessing skill and providing automatic feedback, we believe the multi-task approach suggested in this study will be beneficial when analyzing data for more complex simulation or even for the real operating room.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11548-022-02559-6.

Acknowledgements Funding for this study was provided by the National Institutes of Health grant 1F32EB017084-01 entitled "Automated Performance Assessment System: A New Era in Surgical Skills Assessment."

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval Study approval was granted by the University of Wisconsin Health Sciences Institutional Review Board, and written informed consent was obtained from all participants.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

1. Al Hajj H, Lamard M, Conze PH, Cochener B, Quellec G (2018) Monitoring tool usage in surgery videos using boosted convolutional and recurrent neural networks. Med Image Anal 47:203–218
2. Ali H, Khursheed M, Fatima SK, Shuja SM, Noor S (2019) Object recognition for dental instruments using SSD-MobileNet. In: 2019 international conference on information science and communication technology (ICISCT). IEEE, pp 1–6
3. Campain NJ, Kailavasan M, Chalwe M, Gobeze AA, Teferi G, Lane R, Biyani CS (2018) An evaluation of the role of simulation training for teaching surgical skills in sub-Saharan Africa. World J Surg 42(4):923–929
4. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S (2020) End-to-end object detection with transformers. In: European conference on computer vision. Springer, pp 213–229
5. Cleary K, Chung HY, Mun SK (2004) Or2020 workshop overview: operating room of the future. In: International congress series. Elsevier, vol 1268, pp 847–852
6. D'Angelo ALD, Rutherford DN, Ray RD, Laufer S, Kwan C, Cohen ER, Mason A, Pugh CM (2015) Idle time: an underdeveloped performance metric for assessing surgical skill. Am J Surg 209(4):645–651
7. D'Angelo ALD, Rutherford DN, Ray RD, Laufer S, Mason A, Pugh CM (2016) Working volume: validity evidence for a motion-based metric of surgical efficiency. Am J Surg 211(2):445–450
8. Darzi A, Smith S, Taffinder N (1999) Assessing operative skill: needs to become more objective
9. Davies J, Khatib M, Bello F (2013) Open surgical simulation—a review. J Surg Educ 70(5):618–627
10. Du X, Kurmann T, Chang PL, Allan M, Ourselin S, Sznitman R, Kelly JD, Stoyanov D (2018) Articulated multi-instrument 2-D pose estimation using fully convolutional networks. IEEE Trans Med Imaging 37(5):1276–1287
11. Eckert M, Cuadrado D, Steele S, Brown T, Beekley A, Martin M (2010) The changing face of the general surgeon: national and local trends in resident operative experience. Am J Surg 199(5):652–656
12. Fard MJ, Ameri S, Chinnam RB, Pandya AK, Klein MD, Ellis RD (2016) Machine learning approach for skill evaluation in robotic-assisted surgery. In: Proceedings of the world congress on engineering and computer science, vol 1
13. Fawaz HI, Forestier G, Weber J, Idoumghar L, Muller PA (2018) Evaluating surgical skills from kinematic data using convolutional neural networks. In: International conference on medical image

computing and computer-assisted intervention. Springer, pp 214–221
14. Fonseca AL, Evans LV, Gusberg RJ (2013) Open surgical simulation in residency training: a review of its status and a case for its incorporation. J Surg Educ 70(1):129–137
15. Funke I, Mees ST, Weitz J, Speidel S (2019) Video-based surgical skill assessment using 3D convolutional neural networks. Int J Comput Assist Radiol Surg 14(7):1217–1225
16. Gallagher AG, O'Sullivan GC (2011) Fundamentals of surgical simulation: principles and practice. Springer, New York
17. Gao M, Bai Y, Li Z, Li S, Zhang B, Chang Q (2021) Real-time jellyfish classification and detection based on improved YOLOV3 algorithm. Sensors 21(23):8160
18. Gao Y, Vedula SS, Reiley CE, Ahmidi N, Varadarajan B, Lin HC, Tao L, Zappella L, Béjar B, Yuh DD, Chen CCG, Vidal R, Khudanpur S, Hager G (2014) JHU-ISI gesture and skill assessment working set (JIGSAWS): a surgical activity dataset for human motion modeling. In: MICCAI workshop: M2cai, vol 3, p 3
19. Gao Y, Vedula SS, Reiley CE, Ahmidi N, Varadarajan B, Lin HC, Tao L, Zappella L, Béjar B, Yuh DD, Chen CCG, Vidal R, Khudanpur S, Hager G (2014) Language of surgery: A surgical gesture dataset for human motion modeling. Modeling and monitoring of computer assisted interventions (M2CAI), 2014
20. García Vazquez A, Verde JM, Dal Mas F, Palermo M, Cobianchi L, Marescaux J, Gallix B, Dallemagne B, Perretta S, Gimenez ME (2020) Image-guided surgical e-learning in the post-Covid-19 pandemic era: what is next? J Laparoendosc Adv Surg Tech 30(9):993–997
21. Genc V, Sulaimanov M, Cipe G, Basceken SI, Erverdi N, Gurel M, Aras N, Hazinedaroglu SM (2011) What necessitates the conversion to open cholecystectomy? A retrospective analysis of 5164 consecutive laparoscopic operations. Clinics 66(3):417–420
22. Goldbraikh A, D'Angelo ALD, Pugh CM, Laufer S (2020) Tool usage in open surgery video data. In: Computer assited radialogy and surgery (CARS) 2020
23. Hasan OH, Ayaz A, Khan M, Docherty C, Hashmi P (2019) The need for simulation in surgical education in developing countries. The wind of change. Review article. J Pak Med Assoc 69(Supl. 1):S62
24. Hassan NI, Tahir NM, Zaman FHK, Hashim H (2020) People detection system using YOLOV3 algorithm. In: 2020 10th IEEE international conference on control system, computing and engineering (ICCSCE). IEEE, pp 131–136
25. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask R-CNN. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969
26. Herzig R, Levi E, Xu H, Gao H, Brosh E, Wang X, Globerson A, Darrell T (2019) Spatio-temporal action graph networks. In: Proceedings of the IEEE/CVF international conference on computer vision workshops
27. Hu X, Yu L, Chen H, Qin J, Heng PA (2017) Agnet: attention-guided network for surgical tool presence detection. In: Deep learning in medical image analysis and multimodal learning for clinical decision support. Springer, pp 186–194
28. Jin A, Yeung S, Jopling J, Krause J, Azagury D, Milstein A, Fei-Fei L (2018) Tool detection and operative skill assessment in surgical videos using region-based convolutional neural networks. In: 2018 IEEE winter conference on applications of computer vision (WACV). IEEE, pp 691–699
29. Jo K, Choi Y, Choi J, Chung JW (2019) Robust real-time detection of laparoscopic instruments in robot surgery using convolutional neural networks with motion vector prediction. Appl Sci 9(14):2865
30. Katić D, Wekerle AL, Gärtner F, Kenngott H, Müller-Stich BP, Dillmann R, Speidel S (2014) Knowledge-driven formalization of laparoscopic surgeries for rule-based intraoperative context-aware assistance. In: International conference on information processing in computer-assisted interventions. Springer, pp 158–167
31. Lewis LL, Kerna NA (2019) Cognitive apprenticeship appropriate surgical education for countries with limited resources. Surg Med Open Access J
32. Li Y, Zhao Z, Luo Y, Qiu Z (2020) Real-time pattern-recognition of GPR images with YOLO V3 implemented by TensorFlow. Sensors 20(22):6476
33. Li YL, Zhou S, Huang X, Xu L, Ma Z, Fang HS, Wang Y, Lu C (2019) Transferable interactiveness knowledge for human-object interaction detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3585–3594
34. Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp 2980–2988
35. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) SSD: single shot multibox detector. In: European conference on computer vision. Springer, pp 21–37
36. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: hierarchical vision transformer using shifted windows. In: International conference on computer vision (ICCV)
37. Luck J, Gosling N, Saour S (2021) Undergraduate surgical education during Covid-19: could augmented reality provide a solution? Br J Surg 108(3):e129–e130
38. McCoy AC, Gasevic E, Szlabick RE, Sahmoun AE, Sticca RP (2013) Are open abdominal procedures a thing of the past? An analysis of graduating general surgery residents' case logs from 2000 to 2011. J Surg Educ 70(6):683–689
39. Moorthy K, Munz Y, Sarker SK, Darzi A (2003) Objective assessment of technical skills in surgery. BMJ 327(7422):1032–1037
40. Partridge RW, Hughes MA, Brennan PM, Hennessey IA (2014) Accessible laparoscopic instrument tracking ("instrac"): construct validity in a take-home box simulator. J Laparoendosc Adv Surg Tech 24(8):578–583
41. Perez A, Klimberg VS (2021) Guest editorial "tele-education and tele-mentoring". J Surg Oncol 124(2):161–161. https://doi.org/10.1002/jso.26501
42. Ramesh S, Dall'Alba D, Gonzalez C, Yu T, Mascagni P, Mutter D, Marescaux J, Fiorini P, Padoy N (2021) Multi-task temporal convolutional networks for joint recognition of surgical phases and steps in gastric bypass procedures. Int J Comput Assist Radiol Surg 1–9
43. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
44. Redmon J, Farhadi A (2017) Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7263–7271
45. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767
46. Reiley CE, Lin HC, Yuh DD, Hager GD (2011) Review of methods for objective surgical skill evaluation. Surg Endosc 25(2):356–366
47. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst 28:91–99
48. Reynolds W Jr (2001) The first laparoscopic cholecystectomy. JSLS J Soc Laparoendosc Surg 5(1):89
49. Reznick RK, MacRae H (2006) Teaching surgical skills-changes in the wind. N Engl J Med 355(25):2664–2669
50. Roach E, Okrainec A (2021) Telesimulation for remote simulation and assessment. J Surg Oncol 124(2):193–199. https://doi.org/10.1002/jso.26505
51. Saun TJ, Zuo KJ, Grantcharov TP (2019) Video technologies for recording open surgery: a systematic review. Surg Innov 26(5):599–612

52. Schroder M, Ritter H (2017) Hand-object interaction detection with fully convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 18–25
53. Siddiqui UD, Aslanian HR (2020) The new virtual reality: advanced endoscopy education in the Covid-19 era. Dig Dis Sci 65:1888–1891
54. Soviany P, Ionescu RT (2018) Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction. In: 2018 20th international symposium on symbolic and numeric algorithms for scientific computing (SYNASC). IEEE, pp 209–214
55. Twinanda AP, Alkan EO, Gangi A, de Mathelin M, Padoy N (2015) Data-driven spatio-temporal RGBD feature encoding for action recognition in operating rooms. Int J Comput Assist Radiol Surg 10(6):737–747
56. Twinanda AP, Shehata S, Mutter D, Marescaux J, De Mathelin M, Padoy N (2016) EndoNet: a deep architecture for recognition tasks on laparoscopic videos. IEEE Trans Med Imaging 36(1):86–97
57. VanVoorst BR, Hackett M, Strayhorn C, Norfleet J, Honold E, Walczak N, Schewe J (2015) Fusion of lidar and video cameras to augment medical training and assessment. In: 2015 IEEE international conference on multisensor fusion and integration for intelligent systems (MFI). IEEE, pp 345–350
58. Yadav SK, Mishra A, Mishra SK (2021) Telemedicine: history and success story of remote surgical education in India. Indian J Surg 1–5
59. Yoon Y, Hwang H, Choi Y, Joo M, Oh H, Park I, Lee KH, Hwang JH (2019) Analyzing basketball movements and pass relationships using realtime object tracking techniques based on deep learning. IEEE Access 7:56564–56576
60. Zhang M, Cheng X, Copeland D, Desai A, Guan MY, Brat GA, Yeung S (2020) Using computer vision to automate hand detection and tracking of surgeon movements in videos of open surgery. In: AMIA annual symposium proceedings. American Medical Informatics Association, vol 2020, p 1373
61. Zia A, Essa I (2018) Automated surgical skill assessment in RMIS training. Int J Comput Assist Radiol Surg 13(5):731–739