SINGLE SHOT STATE DETECTION IN SIMULATION-BASED LAPAROSCOPY TRAINING

Kuo Shiuan Peng
Minsik Hong
Jerzy Rozenblit

Department of Electrical
and Computer Engineering
The University of Arizona
1230 E Speedway Blvd.
Tucson, AZ, USA
{kspeng, mshong, jerzyr}@email.arizona.edu

Allan J. Hamilton

Department of Surgery
The University of Arizona
1501 N. Campbell Ave.
Tucson, AZ, USA
allan@surgery.arizona.edu

ABSTRACT

A Single Shot State Detection (SSSD) method is proposed to support a laparoscopic surgery skills training system – Computer-Assisted Surgical Trainer (CAST). CAST actively assists a trainee with visual, audio, or force guidance during different surgical practice tasks. In each task, the guidance is provided according to the target object state, which is one of the key components of CAST. We propose SSSD using deep neural networks to detect object states in a single image. We first model semantic objects to recognize objects' state given a training task and then apply a deep learning algorithm, single shot detector (SSD), to detect the semantic objects. The contribution of this research is to present a unified object state model collaborating with a deep learning object detector, which can be applied to the surgical training simulator, as well as other visual sensing and automation systems.

Keywords: object state detection, semantic object, laparoscopic surgery training.

1 INTRODUCTION

1.1 Background

Laparoscopic surgery is a popular and advanced technique, which offers patients the benefits of minimal invasiveness and fast recovery time. However, one of the main challenges of this technique is the hand-eye coordination using the monocular endoscopic camera and the long, thin special surgical instruments. Before the operation on patients, medical students and residents need to take extensive simulation-based training of laparoscopic surgical training tasks. Currently, the surgical simulators are categorized into two groups: low fidelity and high fidelity (Hammoud et al. 2008). The fidelity of a simulator is defined by the extent to which it provides realism through characteristics, such as visual cues, tactile sensations, feedback capability, and the interaction with a trainee. The most common and inexpensive simulators, the Video Box Trainers (Scott et al. 2000), are using real surgical instruments and a box with slits on the anterior surface for trocar insertion, which has limited feedback. In addition, the assessment method of the Video Box Trainers is to hire a human trainer to directly observe the operation and then the human trainer provides the evaluation

result. To develop a high fidelity simulator, an intelligent simulator with multiple aspects of feedback, and an automatic and objective assessment method is important in the surgical simulator development.

1.2 Problem Statement

Computer-Assisted Surgical Trainer (CAST) (Rozenblit et al. 2014) is a simulation-based training system designed for the laparoscopic surgery skill training. CAST can provide visual-, audio-, force-guidance, and objective assessment metrics for a trainee in a training task, e.g., Peg Transfer Task (PTT). PTT is the one of hands-on exams of Fundamentals of Laparoscopic Surgery (FLS) program (2018), developed by the Society of American Gastrointestinal and Endoscopic Surgeons (SAGES). This program has multiple tests for trainees to learn the basic laparoscopic skills. CAST aims to support all the tasks of the exams to assist a trainee to master these skills.

To achieve this goal, CAST is being developed to provide several kinds of the guidance according to the object states in the training. The object state is defined as the objects' location and interactions among objects. For instance, an audio guidance provides audio instructions to lead a trainee to operate an instrument to grasp a target object based on where the target object and the instrument are and what the interaction between them is. Hence, the object state in a task is the foundation of the guidance in CAST. All the guidance is based on the detected object states to provide active or passive guidance, such as overlaying visual cues, audio instructions, and force feedback. Then the assessment agent in CAST evaluates the performance of the trainee and presents the assessment results. In short, the object state detection method is the key of the information processing and feedback generation in CAST.

1.3 Problem Formulation and Research Structure

An object state detection method has two fundamental steps: object state modeling and detection. Each model would vary due to the unique configurations of different applications, i.e., one single detector is unlikely to support all kinds of the training tasks in CAST. In this research, we aim to design a common methodology to model the object state to have a universal state detection method for all the training tasks. The key to achieving this goal is to convert the object state recognition to an object detection problem. First, the object state is modeled as a semantic object, which contains the target object class and the interaction with objects simultaneously. Then a deep learning based object detection method is applied to locate and detect the designed semantic objects in an image. In our design, it does not matter to choose which kind of object detection method. However, the performance of our model highly relies on the object recognition ability of the selected detection method. In this paper, the current state-of-the-art deep learning object detector, single shot detection (SSD) (Liu et al. 2016), is adopted. Based on our design, each object state, including location and the interaction with others, in different tasks can be converted into the same type of the semantic object. Hence, a single object detection method can unify the object state detection process for different tasks. We name this method as Single Shot State Detection (SSSD).

In summary, this research includes two goals: a) the semantic object modeling for the training task of the CAST system, b) the semantic object detection using deep learning to effectively support real-time application of the CAST system. The main contributions of this research are to present an unified object state modeling and a new application of the existing object detection method. The proposed method can be applied to not only the surgical training simulator but also any other visual sensing and automation systems, such as intelligent surveillance systems or autonomous driving systems.

The content of this paper is structured as the following order: 1) the recent research of semantic objects and object detection are reviewed in the Related Works section, 2) the proposed object state modeling

method and the utilized detection model are elaborated in the Methodology section, 3) the performance of the proposed method is evaluated in the Simulation section, and 4) the conclusion is presented in the last section.

2 RELATED WORKS

In this paper, the object state detection is formulated as the semantic object detection. The related works of the semantic object detection and the object detection are reviewed in this section.

A semantic object is defined as a representation of a collection of attributes that describes an identifiable object in the environment (Benson 2000). To detect a semantic object is to interpret the scene semantics. A large body of work has focused on semantic object segmentation, parsing, and labeling, using either single image or video images. In this paper, we only focus on the single image approaches. The individual objects are detected first, and their relationships are then inferred using geometrical model matching (Guenther et al. 2017), Local-Global Long Short Term Memory spatial dependency (Liang et al. 2016), or Graph Topology (Liang et al. 2016). This approach relies on the spatial relationships among objects to generate the semantics to label the objects. Another group of researches is based on conditional random field (CRF) to infer the object semantics (Zheng et al. 2015). This approach still needs to detect the objects first and then build a factor graph among the detected objects. The two step process, object detection and association, needs high computational cost to apply to the real-time task. An approach to directly detect semantic object without the object association process in the real-time manner is missing.

Several high accuracy and efficient modern convolutional neural network (CNN) based object detectors are devised (Zhou et al. 2014) because of the emerging deep learning techniques. Regional-based Fully Convolutional Network (RFCN) (Dai and Li 2016) can find the possible region of an object in an image first and then classify the object in the proposed region. RFCN also utilizes a multi-scale detector in the detection process to obtain accurate and scale-independent detection results. However, the computational efficiency of RFCN is constrained by the region proposal process. While an alternative method, You Only Look Once (YOLO), was proposed as a multibox detector to identify all the possible objects in a single shot (Redmon et al. 2016), it still suffers from the scale constraint. To solve this problem, single shot detection (SSD) utilizes the multibox and multi-scale method simultaneously to detect objects in real-time manner without any constraint (Liu 2016).

Previous works have demonstrated remarkable results in semantic object parsing and object detection, but none of them can support the real-time application of the object state detection. In this paper, a single shot state detection (SSSD) method is designed to achieve the object semantics interpretation. With the powerful feature extraction ability of the deep convolutional network, the object semantics can be estimated from a single image. The main contributions of this work include 1) a universal semantic object modeling approach to represent object states, and 2) a new application of the existing object detection using deep convolutional network. The key advantage of the proposed SSSD method is that the semantic object model can eliminate the object association process in the detection.

3 METHODOLOGY

3.1 Experimental Task

To verify the design of the proposed SSSD method, a training task is implemented to evaluate the performance. We choose PTT, the first hands-on exam in the FLS program, to be our experimental task. It includes two surgical instruments (graspers) from left and right side, one pegboard with twelve pegs, and six rubber ring-like objects (triangles) as shown in Figure 1. A trainee is asked to control a grasper to lift a triangle

from a peg, carry this object to midair, transfer it to the other grasper, and place the triangle on a peg on the opposite side of the pegboard. If the triangle is dropped during the procedure, the task is terminated and the trainee fails this exam. In this task, trainees learn how to manipulate grasper-type instruments and then improve their eye-hand coordination, ambidexterity, and depth perception with the monocular images.

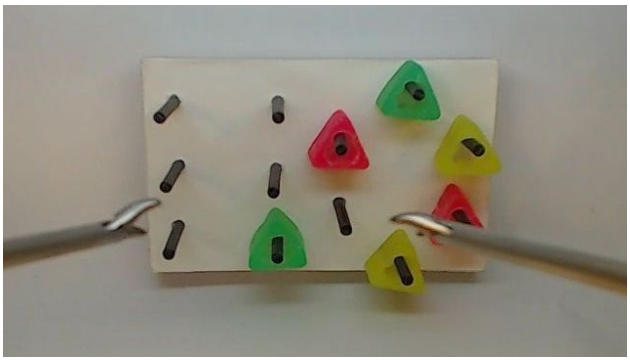


Figure 1: Peg transfer task

3.2 Object State Modeling

An object state model describes the attributes of the target object in the task. We firstly define the target objects of the task. A trainee should focus on using graspers to control a triangle without dropping it. We only need to focus on the states of the triangles and graspers, our target objects. The second part of the model is the definition of the target object states, consisting of two main attributes: *location* and *activity*.

The first attribute, *location*, is the pixel location $l_i = (cx, cy)_i$ of the i^{th} target object ($o_i \in O$, where O is a set of objects) center in the image. *Activity* ($a_i \in A$, where A is a set of activities), the later attribute, is the interaction among a target object i and any other objects. For example, a triangle is transferred by a left grasper to the right one and "being transferred" is the current activity of the triangle object. Therefore, each target object may have multiple activity attributes. A commonly used state of o_i can be presented as $s_i = (l_i, \mathbf{a_i})$, where $\mathbf{a_i} = (a_1, a_2, ...a_j)_i$ and j represent the number of activities. The $activity$ set A of PTT is defined in Table 1. There are some overlapped activities between the triangle and graspers. For example, the activity "Connect" of a grasper is implicitly a part of the activities "Pick-Place" and "Carry" of a triangle, while "Free" is a part of the rest activities of the triangle. In other words, the activity attributes of a grasper can be inferred from those of a triangle, and we only need to focus on grasper's location. Only a triangle object has the activity and location in its state.

To eliminate this inconsistency of the object state model, we propose a semantic object state model to simplify the state definition shown in Table . As a new type of object, semantic object merges an activity attribute with its object class and then each semantic object ($q_i \in Q$, where Q is a set of semantic objects) has only one attribute, location (i.e., o_i can be represented as q_i, which has l_i).

The advantage of the proposed semantic object is to simplify an object state detection to be an object detection problem. Instead of detecting the *activity* and *location* attributes of the target objects, we only need to detect the location of the defined semantic objects.

3.3 ingle Shot Detection

After modeling the semantic objects, a semantic object detector is presented in this section. The state-of-the-art single shot detector, SSD, designed by Liu et al. (2016) is used to achieve our goal in the proposed

Table 1: Object activity definition table.

Target Object	**Activity**
Triangle	On Peg:A triangle is not connected to any grasper but on a peg. Left Pick-Put: A triangle is connected to the left grasper and also on a peg or on the ground. Right Pick-Put: A triangle is connected to right grasper and also on a peg or on the ground. Left Carry: A triangle is connected to the left grasper but not on any peg. Right Carry: A triangle is connected to the right grasper but not on any peg. Transfer: A triangle is being transferred from one grasper to another but not on any peg. Out Peg: A triangle is not connecting to any grasper and not on any peg or on the ground.
Left Grasper	Free: The left grasper is not connecting to any triangle. Connect: The left grasper is connecting to a triangle.
Right Grasper	Free: The right grasper is not connecting to any triangle. Connect: The right grasper is connecting to a triangle.

Table 2: Semantic object model of PTT.

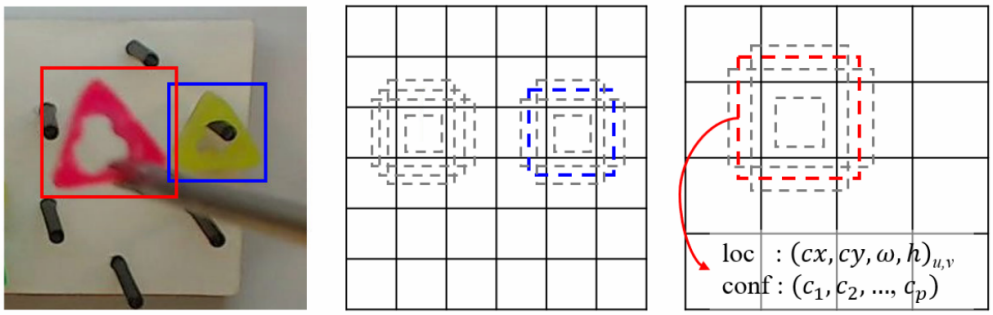
#	**Semantic Objects**	**Alias**
1	On-Peg-Triangle	onpeg
2	Left-Pick-Place-Triangle	l-pick
3	Right-Pick-Place-Triangle	r-pick
4	Left-Carry-Triangle	l-carry
5	Right-Carry-Triangle	r-carry
6	Transfer-Triangle	transfer
7	Out-Peg-Triangle	outpeg
8	Left-Grasper	l-grasp
9	Right-Grasper	r-grasp

method. In this section, the framework of SSD is reviewed. The main design concept - Multiscale Bounding Box Prediction - and the training configuration are explained.

3.3.1 Multiscale Bounding Box Prediction

The target output of the SSD approach is to predict the presence of an object in different scales. The method utilized in the algorithm is named Multiscale Bounding Box Prediction. The framework is shown in Figure 2. First, an input image with a labeled ground truth box (GTB) for each object shown in Figure 2(a) is fed into the feature learning network. The learning network of the SSD model is based on the backbone of a standard CNN architecture designed for the semantic object classifier. The selected CNN network is VGG-16 (Simonyan and Zisserman 2014), the architecture of a high quality image classifier (Top place of Classification Competition in ImageNet ILSVRC-2014) devised by Visual Geometry Group. In the learning network, the output is evaluated as to multiple fixed scale grid feature maps, indicated by solid lines, and a set of default boxes with different aspect ratios, indicated by dash lines, at each location, where two exemplar feature maps 6×6 and 4×4 are shown in Figure 2(b) and (c).

The different scale feature maps are designed to detect the multiple scale object. In the end, all the feature maps are collected into the final feature map. In each default box ($b_{u,v}$ where u represents an index of the default box and v represents an index of the cell) of each cell in the feature map, the shape offsets $so_{u,v} = (cx, cy, w, h)_{u,v}$ and the confidence scores $(C_{u,v} = (c_1, c_2..., c_p)_{u,v}$, where there are p number of



(a) Image with GTBs (b) 6 × 6 feature map (c) 4 × 4 feature map

Figure 2: SSD Multiscale Bounding Box Prediction framework.

semantic objects) for all object classes are predicted. The shape offset contains the center offset (cx, cy) of the object and the scale ratio of the width (w) and height (h) of the default bounding box.

3.3.2 Training Flow

To successfully train the SSD network to learn the objects, a prepared training set should be ready first and the loss function and back propagation then can be applied. Overall, the training process is based on the default box selection, and the quality of the model highly depends on the dataset augmentation.

First, the ground truth information has to to be assigned in the training dataset and to match the designed output maps format. The labeling process is simply defining the bounding box and the object class of the target objects in the training images. Once the assignment is done, the label information of the images is converted to the output format of the SSD model following the selected default boxes and the corresponding feature map scale. To determine the default boxes corresponding to a ground truth box, we match all the default box in different scales and find the one with best jaccard overlap (Erhan et al. 2014), which is confined over a threshold (> 0.5) to reduce the candidates and simplify the learning problem.

4 SIMULATION

4.1 Task and Metrics

In the simulation, the evaluation is based on the collected dataset of PTT, named CAST PTT dataset, which is collected through the CAST system. Two trials of the operation are proceeded and recorded as in the CAST PTT dataset. Each operation is a complete flow to transfer all the six triangles from one side to the other. There are in total 5000 sample images extracted from the two recorded videos. The training set has randomly selected 4000 images, while the testing set has the rest 1000. The statistics of the semantic object amount is listed in Table 3. The onpeg has the highest amount in each set, because only one triangle is allowed to be moved in every single operation and the rest of the five triangles remain on the pegs. To the contrary, the least semantic object is the outpeg due to the rapid movement of a dropping triangle. The l- and r-graspers have the similar amount of the count as the captured frames. Only in few cases the l- and r-graspers have different counts as the captured frames, because the tip of the grasper is out of the scene and it is not possible to label the out-of-scene grapser.

To evaluate the proposed method, we first calculate the confusion matrix for the semantic object classification. We then compute the mean and standard deviation of the Intersection of the Union (IoU) and the error

Table 3: PTT semantic objects statistics in training and test sets.

Semantic Object	Training Set Count	Test Set Count
onpeg	19942	4995
transfer	902	228
outpeg	371	94
l-pick	376	108
r-pick	220	53
l-carry	904	226
r-carry	300	183
l-grasp	4000	1000
r-grasp	3967	995
total	30982	7882

distance (d_{err}) between the ground truth and the prediction respectively for each semantic object to verify the accuracy of the location detection.

The IoU is the intersection of the union between the ground truth and the detected bounding boxes. the definition of IoU is as:

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}.$$

The higher IoU indicates better detection accuracy. The d_{err} is the difference between the centers of the ground truth and the detected bounding boxes, which can help measure the localization result for each semantic object.

4.2 Simulation Configuration

To design a deep learning algorithm for the CAST system, Python and Tensorflow in Linux Ubuntu system are used to develop the prototype under the GPU computing setting. A common NVIDIA GTX 1080 graphic card is used in the development and simulation to ensure the algorithm design is feasible in the personal computing environment. All input images are resized to a 300×300 from the original size, e.g. 1280×800, of the training image and the model fine-tune on the pretrained VGG-16 (Simonyan and Zisserman 2014). The predictions happen in less than 17 milliseconds (around 60 frames per second), which is fast and feasible for real-time applications.

4.3 Results

The quantitative simulation results are shown in Table 4 and Table 5. The Table 4 is the confusion matrix of the classification detection results, while the Table 5 indicates the error of the localization detection result. The example of the visual simulation result is shown in Figure 3. The full sequence of the simulation was recorded as video and available in the following URL: https://youtu.be/9gPfhdBiwoE.

In Table 4, the **onpeg**, **outpeg**, **l-grasp**, and **r-grasp** have both over 90% accuracy and precision. These are static semantic objects and the detection process is similar to the conventional object detection, so the detection results are stable and accurate. The precision of the **l-pick** and **l-carry** are under 80% and lower than the ones of the **r-pick** and **r-carry**. However, the total amount of the **l-pick** and **l-carry** are higher

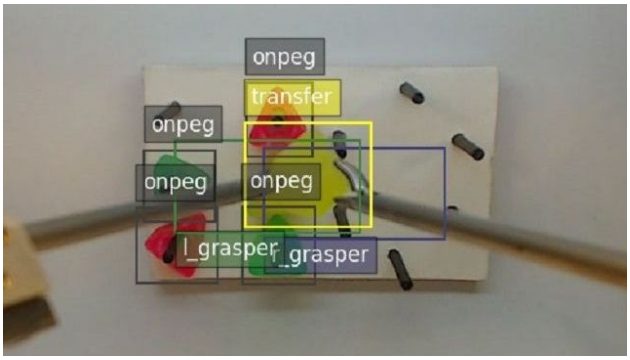


Figure 3: Simulation result.

than the ones of the **r-pick** and **r-carry**. It indicates that the left-hand movements are more than the right-hand and the detection of the left side semantic objects is also less precise. This result matches the testing scenario, which the operator of PTT is right-handed. The right-handed operator may take longer time and be less stable while proceeding in the left-hand movements in the task take longer time and less stable. Therefore, the left side cases may have more samples and the labels of the semantic objects may be not very accurate in the case of the transition between **onpeg** and **l-pick**.

Table 4: Confusion matrix of the semantic object detection.

		Prediction									
		onpeg	**transfer**	**outpeg**	**l-pick**	**r-pick**	**l-carry**	**r-carry**	**l-grasp**	**r-grasp**	**Accuracy**
Ground Truth	**onpeg**	4881	4	0	64	12	4	2	2	24	97.76
	transfer	2	167	0	0	0	42	17	0	0	73.25
	outpeg	0	0	86	0	0	4	3	0	0	92.47
	l-pick	2	0	0	105	0	1	1	0	0	96.33
	r-pick	1	0	0	0	49	0	2	0	0	94.23
	l-carry	0	10	3	1	0	210	1	1	0	92.92
	r-carry	1	0	5	0	0	15	161	0	1	87.98
	l-grasp	49	0	0	0	0	0	1	946	5	94.51
	r-grasp	0	1	2	0	0	0	1	12	980	98.39
	Precision	98.87	91.45	90	61.7	89.67	75.72	84.13	98.38	97.03	

Further, the precision of the **l-pick** case is much lower than other cases and the false detection is mainly on the **onpeg** case. The similar situation happens on the **r-pick** case. The possible reason is that the labels of the **onpeg** and **pick** cases are not accurate due to the vague transition between the **onpeg** and **pick** cases in a 2D image. In both the visual and computational process, it is difficult to identify the object state (semantic object) without the 3D location of the grasper and the triangle when the grasper is approaching the triangle. The performance of the detection between the **onpeg** and **pick** cases is limited although the object detection algorithm works well on all the other cases. The possible solution is to involve the 3D information, such as stereo image pairs or depth information to improve the accuracy of the labels and the detection.

In Table 5, the evaluation metrics of the localization are presented. The IoU represents the coverage percentage of the prediction bounding box on the ground truth bounding box, and the higher IoU value indicates a better alignment result of the bounding box. The standard deviation (std) of IoU indicates the variation of the IoU and the smaller std is better. All the cases in the evaluation have over 80% IoU on average. The **onpeg** case has the highest $IoU\ mean$ at 90% because **onpeg** is relatively simple and stable. To further elaborate, the bounding box sizes of all the **onpeg** cases are similar to yield a simple learning target, and training set size of **onpeg** is the largest one that can help the network learn the features more effectively. However, **onpeg** also has largest std because the the the bounding box size of **onpeg** is the smallest one

among all the semantic objects. The triangle in the **onpeg** case is at the farthest location among all cases. The rest of the semantic objects have similar performance.

Table 5: Localization detection evaluation.

Semantic Object	**IoU**(%)		d_{err}**(pixel)**	
	mean	**std**	**mean**	**std**
onpeg	0.92	0.09	2.17	7.73
transfer	0.83	0.07	5.56	3.88
outpeg	0.86	0.06	3.09	2.85
l-pick	0.86	0.04	2.28	1.45
r-pick	0.85	0.06	3.12	1.74
l-carry	0.84	0.07	5.44	4.66
r-carry	0.85	0.06	5.12	4.08
l-grasp	0.85	0.07	6.83	4.73
r-grasp	0.86	0.07	7.04	5.91
total	0.89	0.09	3.66	7.16

The d_{err} is the distance in pixel scale between the prediction bounding box on the ground truth bounding box. The smaller d_{err} shows a more accurate prediction of the bounding box center. Similarly, the **onpeg** case shows the low d_{err} $mean$ at ($\approx$ 2.2 pixels) and high std ($\approx$ 7.7) with the same reasons as the IoU. Surprisingly, the **outpeg**, **l-pick**, and **r-pick** cases have low d_{err} $mean$ ($2.3 \sim 3.1$ pixels) and lower std ($1.5 \sim 2.8$). The **outpeg** case is relatively simple because the triangle in this case has no interaction with any other objects. On the other hand, the **l-** and **r- pick** cases have relatively smaller bounding boxes and the interaction among the triangle, peg, and grasper is clear and relatively more significant. The rest of the semantic objects, including **outpeg**, **l-carry**, **r-carry**, **l-grasp**, and **r-grasp**, which are closer to the camera and have larger bounding boxes, have the higher d_{err} $mean$ ($\approx$ 6 pixels) and lower std (≈ 4.5).

5 DISCUSSION AND FUTURE WORK

Overall, the semantic objects can be detected effectively using SSSD method and the average accuracy is over 90%. The left-handed related semantic objects and the **pick** case perform significantly worse than others. The left-handed related semantic objects suffer from the right-handed operator of the trial and the countermeasure is to employee the expert, who can operate the right- and left-handed objects without significant difference, to proceed to sample trial. The **pick** case has the physical constraint from the 2D images and it is impossible to identify the **onpeg** and **pick** cases with 100% certainty when they are very close visually or computationally. The only way to solve this problem is to introduce the 3D information, such as the stereo image pairs or the depth image, to assist the labeling, learning, and prediction process.

However, SSSD method can only support the 2D image and the performance of our model suffered in some specific cases, which need the 3D location information of each object. Therefore, we also plan to expand the SSSD method to support 3D information but still keep using the single 2D image input in our future work. The significance for simulation-based surgical training is in the ability to recognize in real-time the state of a task (and its objects) so that we could introduce computer-based assistance in completing an exercise with erroneous actions. For instance, when a trainee drops a triangle in the peg transfer task, haptic and visual assistance can be provided to quickly correct this error based on the SSSD. The potential for using this method in the operating theatre is strong as well in that it could assist the surgeons in case of an unexpected event during a procedure. To validate the performance of SSSD on other setups, we are working on preparing experiment for other training tasks. The ultimate goal is to extend our method to support all the

FLS training tasks. Eventually, we will generalize SSSD to other applications and dataset, such as Human Object Interaction Dataset (Chao et al. 2015).

On the other hand, the object detection method is the key to successfully detect the object states. The detection methods can detect semantic objects when they have better ability to extract object features and make associates among these features. In this paper, we directly adopt SSD, which is one of the method with best balance between the accuracy and computational efficiency (Huang et al. 2017). There are other options which can be good candidates to perform SSSD, such as RFCN and YOLO. Thus, the further evaluation of different algorithms on SSSD is planed in the next step. Second, there are also different backbone, e.g., RESNET (Deng et al. 2009), Inception (Szegedy et al. 2015), Mobilenet (Howard et al. 2017), etc., which can be applied in the base network to improve the performance. We will also evaluate these alternates as a future work.

CAST software is being developed using C++ programming language under windows. We aim to support CAST in both GPU- and CPU-based computing environment. Thus, we are evaluating a feasibility to use TensorFlow's C++ API under Windows configuration with Intel i7-6700 CPU and NIVIDA GTX 1080 GPU. However, unlike our simulation setup, the prediction takes about 80 msec using only CPU (12.5 PFS) and 20 msec using GPU (50 FPS) with C++ API. The computation efficiency of CPU-only configuration needs to be improved to support the real-time processing, e.g. processing rate > 30 FPS, in CAST. Hence, we are working on developing a lighter weight deep learning method to reduce the computation and fit our system configuration. Eventually, SSSD will be able to support real-time processing under both CPU- and GPU-based computing system within the CAST system.

6 CONCLUSION

A novel model - Single Shot State Detection (SSSD) - is presented in this paper to recognize the object state to support a laparoscopic surgery skills training system, CAST. The novelty of this paper is that we proposed a semantic object model to replace the object state model and convert the object state detection to an object detection problem. This idea significantly reduces the complexity of the object interaction detection algorithm in computer vision. Based on the current state-of-the-art object detector SSD, we successfully achieve over 90% accuracy in out test task PTT. The computational efficiency of the proposed method is over 60 FPS to support the most real-time applications. This method is feasible to be applied to any Human-Object-Interaction task. In the next step, we plan to generalize the semantic object model to other FLS or other training tasks and consider the different hardware configuration to support more applications.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant Number 1622589 "Computer Guided Laparoscopy Training". Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

Benson, K. C. 2000. *Database Processing Fundamentals, Design, and Implementation.* Mathematics and Computer Education 34, no. 1. Pearson.

Chao, Y. W., Z. Wang, Y. He, J. Wang, and J. Deng. 2015. "Hico: A benchmark for recognizing human-object interactions in images." In Proceedings of the IEEE International Conference on Computer Vision, pp. 1017-1025..

Deng, J., W. Dong, R. Socher, L. J. Li, K. Li, and F. F. Li. 2009. "Imagenet: A large-scale hierarchical image database." In Computer Vision and Pattern Recognition, CVPR 2009. IEEE Conference on, pp. 248–255.

Erhan, D., C. Szegedy, A. Toshev, and D. Anguelov. 2014. "Scalable object detection using deep neural networks." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2147-2154.

FLS 2018. "Fundamentals of Laparoscopic Surgery". https://www.flsprogram.org. Accessed October 14, 2018.

Guenther, M, T. Wiemann, S. Albrecht, and J. Hertzberg. 2017. "Model-based furniture recognition for building semantic object maps." Artificial Intelligence 247. pp. 336-351.

Hammoud, M. M., F. S. Nuthalapaty, A. R. Goepfert, P. M. Casey, S. Emmons, E. L. Espey, J. M. Kaczmarczyk, N. T. Katz, J. J. Neutens, and E. G. Peskin. 2000. To the point: medical education review of the role of simulators in surgical training. American journal of obstetrics and gynecology 199, no. 4 , pp. 338-343.

Howard, A. G. 2013. "Some improvements on deep convolutional neural network based image classification." arXiv preprint arXiv:1312.5402.

Howard, A. G., M. Zhu, B. Chen, D Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. 2017. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861.

Huang, J., V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer. 2017. "Speed/accuracy trade-offs for modern convolutional object detectors." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7310-7311..

ILSVRC 2018. "ImageNet Large Scale Visual Recognition Competition." http://www.image-net.org/challenges/LSVRC/. Accessed October 6, 2018.

Liang, X, X Shen, J Feng, L Lin, and S Yan. 2016."Semantic object parsing with graph lstm." In European Conference on Computer Vision, pp. 125-143. Springer, Cham.

Liu, W., D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg. 2016. "Ssd: Single shot multibox detector." In European conference on computer vision, pp. 21-37. Springer, Cham.

LSVRC 2014. "Large Scale Visual Recognition Challenge 2014 - ImageNet." http://www.image-net.org/challenges/LSVRC/2014/. Accessed October 6, 2018.

Redmon, J., S. Divvala, R. Girshick, and A. Farhadi. 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.

Rozenblit, J W., C. Feng, M. Riojas, L. Napalkova, A. J. Hamilton, M. Hong, P. Berthet-Rayne et al. 2014. "The computer assisted surgical trainer: design, models, and implementation." In Proceedings of the 2014 Summer Simulation Multiconference, p. 30. Society for Computer Simulation International.

Scott, D. J., P C. Bergen, R. V. Rege, R. Laycock, S. T. Tesfay, R. J. Valentine, D. M. Euhus, D. R. Jeyarajah, W. M. Thompson, and D. B. Jones. 2000. Laparoscopic training on bench models: better and more cost effective than operating room experience? Journal of the American College of Surgeons 191, no. 3, pp. 272-283.

Szegedy, C, W Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. 2015. "Going deeper with convolutions." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9.

Tensorflow. "TensorFlow." https://www.tensorflow.org/. Accessed October 8, 2018.

Xiang, Y., R. Mottaghi, and S. Savarese. 2014. ' "Beyond pascal: A benchmark for 3d object detection in the wild." In Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on, pp. 75-82.

Zheng, S., S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. HS. Torr. 2015. "Conditional random fields as recurrent neural networks." In Proceedings of the IEEE international conference on computer vision, pp. 1529-1537.

Zhou, B., A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. 2014. "Object detectors emerge in deep scene cnns." arXiv preprint arXiv:1412.6856.

AUTHOR BIOGRAPHIES

KUO S. PENG is a Ph.D student of Electrical and Computer Engineering at the University of Arizona. He received a Master of Science degree in Management Information System from the University of Arizona. His research interests lie in image processing, computer vision, and deep learning. His email address is kspeng@email.arizona.edu.

MINSIK HONG is a Ph. D. candidate at the University of Arizona. He received a Master of Science degree in Electrical and Computer Engineering from POSTECH, Republic of Korea. His research interests are robotics, control system, fuzzy theory, and modeling and simulation for medical devices. His email address is mshong@email.arizona.edu.

JERZY W. ROZENBLITis University Distinguished Professor, Raymond J. Oglethorpe Endowed Chair in the Electrical and Computer Engineering (ECE) Department, with a joint appointment as Professor of Surgery in the College of Medicine at the University of Arizona. During his tenure at the University of Arizona, he established the Model-Based Design Laboratory with major projects in design and analysis of complex, computer-based systems, hardware/software codesign, and simulation modeling. He presently serves as Director of the Life-Critical Computing Systems Initiative, a research enterprise intended to improve the reliability and safety of technology in healthcare and life-critical applications. His email address is jr@ece.arizona.edu.

ALLAN J. HAMILTON, MD, FACS is Harvard trained physician, a professor of Neurosurgery at the University of Arizona. Dr. Hamilton was elected a Fellow of the American College of Surgeons in 1994. In 1995, Dr. Hamilton was promoted to Chief of Neurosurgery and became the Chairman of the entire Department of Surgery in 1998. He currently holds a tenured professorship in Neurosurgery, as well as additional professorships in the Departments of Psychology, Radiation Oncology, and the School of Electrical and Computer Engineering. He is Executive Director of the Arizona Simulation Technology and Education Center in the College of Medicine, University of Arizona. His email address is allan@surgery.arizona.edu.