Autonomous Cricothyroid Membrane Detection and Manipulation Using Neural Networks and a Robot Arm for First-Aid Airway Management

Xiaoxue Han
Department of Mechanical Engineering,
Virginia Tech,
Blacksburg, VA 24061
e-mail: hanxiaoxue@vt.edu

Hailin Ren
Department of Mechanical Engineering,
Virginia Tech,
Blacksburg, VA 24061
e-mail: hailin@vt.edu

Jingyuan Qi
Department of Computer Science,
Virginia Tech,
Blacksburg, VA 24061
e-mail: jingyq1@vt.edu

Pinhas Ben-Tzvi[1]
Robotics and Mechatronics Lab,
Department of Mechanical Engineering,
Virginia Tech,
Blacksburg, VA 24061
e-mail: bentzvi@vt.edu

Cricothyrotomy serves as one of the most efficient surgical interventions when a patient is enduring a can't intubate can't oxygenate (CICO) scenario. However, medical background and professional training are required for the provider to establish a patent airway successfully. Motivated by robotics applications in search and rescue, this work focuses on applying artificial intelligence techniques to the precise localization of the incision site, the cricothyroid membrane (CTM), of the injured using an RGB-D camera, and the manipulation of a robot arm with reinforcement learning to reach the detected CTM keypoint. In this paper, we proposed a deep learning-based model, the hybrid neural network (HNNet), to detect the CTM with a success rate of 96.6%, yielding an error of less than 5 mm in real-world coordinates. In addition, a separate neural network was trained to manipulate a robotic arm for reaching a waypoint with an error of less than 5 mm. An integrated system that combines both the perception and the control techniques was built and experimentally validated using a human-size manikin to prove the overall concept of autonomous cricothyrotomy with an RGB-D camera and a robotic manipulator using artificial intelligence. [DOI: 10.1115/1.4056505]

1 Introduction

It is life-critical to establish a patent airway in a timely manner when dealing with an injured with failed airways in life-threatening situations, such as the presence of a foreign body in the airways, angioedema, or massive facial trauma [1]. In these can't intubate can't oxygenate (CICO) scenarios, cricothyrotomy is always regarded as the last resort when orotracheal and nasotracheal intubation is impossible [2]. Also, cricothyrotomy requires less personnel and less equipment as well as introduces lesser tissue dissection; in return, it reduces the overall waiting time for surgery and results in less bleeding, making it a more efficient and safe surgical procedure compared with other commonly used techniques in emergency scenarios [3]. To correctly detect the position of cricothyroid membrane is the first step in successfully performing cricothyrotomy: the patient lays flat on the back and the physician is asked to identify the location of the cricothyroid membrane by locating the intersection between the traverse line and the horizontal line [4], as shown in Fig. 1. In the last few decades, various cricothyrotomy devices have been designed for emergency tracheal intubation with improved performance [5]. However, simulated training is required for the trainees to obtain skills and knowledge to successfully perform the surgery under real-world high-stress situations [6].

This work focuses on developing an integrated system to detect the cricothyroid membrane (CTM) position of an injured person and to control a robotic manipulator to perform the surgery using a needle cricothyrotomy kit. The motivating application behind this work is to deploy robots in search and rescue (SAR) operations with a focus on victim extraction and medical assistance. Although many SAR robots have been developed in the last few decades, most of them focused on the searching tasks, leaving the rescue research still at its initial stage [7]. To address the challenges associated with search and rescue robots on victim extraction, such as human-robot safety concerns, communication setup, and traversability over rough terrains, the Semi-Autonomous Victim Extraction Robot (SAVER) was proposed. Considering cricothyrotomy as a real-time application in search and rescue scenarios, traditional control methods such as remote teleoperation systems controlled by a companion field medic have inevitable drawbacks. Remote operation poses high requirements for the communication setup to provide a good situational awareness and low-latency control for the remote operators [8]. Deploying a field medic in such scenarios could pose great risks to both rescuers and victims [9].

In order to solve the aforementioned problems, while at the same time being inspired by recent work on Vision-Based Control of mobile manipulators [10] and emerging technologies on designing and manufacturing for robotic surgery applications [11], we proposed an autonomous robotic first-aid airway management system that can perform cricothyrotomy on the patient. Perception, decision-making, and control were all embedded in the system. With the real-time high-resolution RGB-D images fed

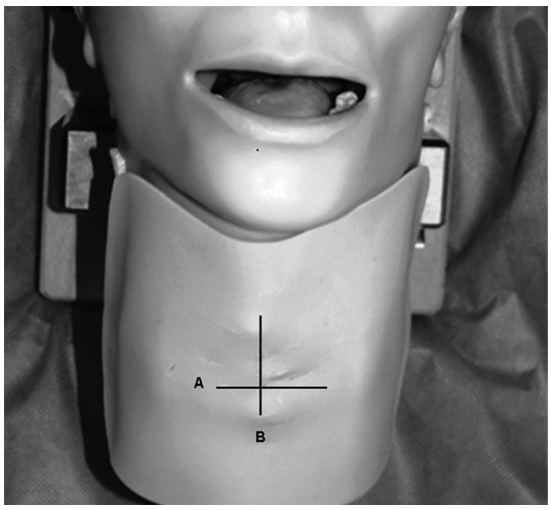

Fig. 1 Image of skin markings (made on a mannequin for illustrative purposes). Line A is the transverse marking and line B is the longitudinal line [4].

[1]Corresponding author.

Manuscript received April 8, 2022; final manuscript received December 5, 2022; published online January 11, 2023. Assoc. Editor: Prasanna Hariharan.

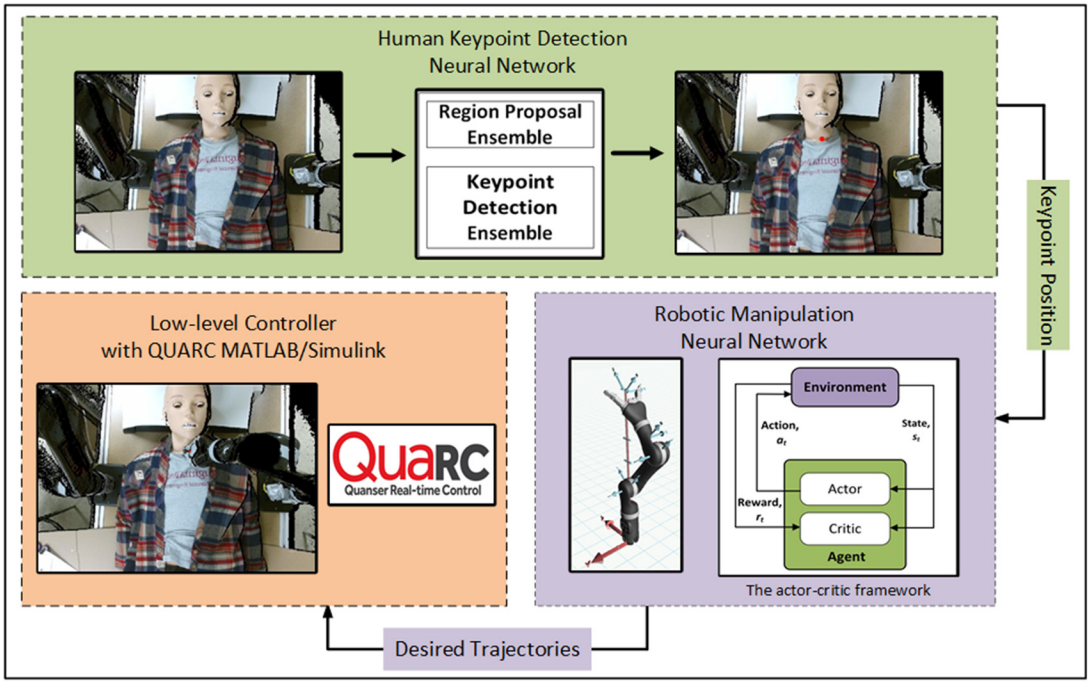

Fig. 2 Integrated CTM detection and manipulation system consisting of the location of the CTM detection, robot arm trajectory estimation, and robot arm control

from the camera, the system could predict the precise location of the incision site for cricothyrotomy. With the position information of the CTM, the robot arm would be manipulated autonomously to complete a series of operations with a commercial cricothyrotomy kit and perform the incision.

The first task is the autonomous detection of the cricothyroid membrane position. In recent years, computer vision with artificial intelligence (AI) technologies has achieved significant developments in a wide range of tasks such as human pose estimation, object detection, facial recognition, etc. [12,13]. With deep learning as a powerful tool, computer vision can provide more delicate performance compared to humans in many scenarios [12].

In our previous work [14], we proposed a hybrid neural network (HNNet) that consists of two ensembles. The first ensemble takes in a compressed image for region-of-interest (ROI) detection; the original high-resolution image is then cropped and fed into the second ensemble, alongside the feature map extracted from the first ensemble, for precise keypoint detection. By doing so, the network generates a prediction on an uncompressed image without involving significantly large computation and satisfies the high-efficiency and high-accuracy requirements.

In this paper, we further improved the performance of the HNNet by introducing more advanced network structures. Also, we manipulated the robot arm to perform a sequence of operations. Learning-based algorithms have drawn much attention in recent years and have proven noteworthy performances on a variety of tasks [15] compared with traditional control mechanisms. We implemented reinforcement learning to teach the robot arm some basic actions, including reaching a specific position and grasping objects. The detection process and robot arm manipulation were incorporated into an integrated perception-control system. In this paper, the system was built based on the assumption that the target was immobilized before and during the whole procedure. The system was realized and tested with a Kinect V2 [16] RGB-D camera and a MICO robot arm manipulator.

The rest of this paper is organized as follows. The proposed methods for both keypoint detection and robot arm manipulation are described in Sec. 2. Section 3 details the results of the proposed method. Section 4 concludes the work with directions for future research.

2 Proposed Algorithm

An autonomous robotic first-aid airway management system is designed to perform Cricothyrotomy on a patient. The system requires both embedded perception and control, as shown in Fig. 2. First, the system detects the precise location of the CTM, the incision site of Cricothyrotomy; then, the trained robot arm manipulation learning neural network would estimate the trajectory for the robot arm. Last, the robot arm is controlled by the low-level controller, QUARC, to follow the trajectory and reach the detected position for subsequent operations.

2.1 Hybrid Neural Network. In our previous work [14], we proposed a HNNet that could balance both the running time and the prediction accuracy for processing high-resolution images. The HNNet consists of a region proposal ensemble and a keypoint detection ensemble. The regional proposal ensemble first takes in the compressed image and selects the ROI. A feature map is extracted to provide spatial information. The original high-resolution image is then cropped according to the selected ROI, and the feature map is also cropped around the ROI with a larger span to provide sufficient spatial information. Both the cropped image and the cropped feature map are fed into the keypoint detection ensemble on different stages to make the final detection. In the previous version of the HNNet, we applied a stacked hourglass network as the keypoint detection model, and it was called HNNet^{HG}.

In this paper, the performance of HNNet^{HG} is further improved with the hybrid architecture remaining the same, but the keypoint detection ensemble being changed. A multi-stage network [12], which consists of several blocks of convolutional layers group followed by a very deep convolutional network with 19 layers (VGG19) [17] model, is chosen for keypoint detection. At each stage, a heatmap is extracted for loss calculation, and it is concatenated with the VGG19 output as the input of the next block. The number of stages is adjusted to 6, 4, and 3, and the corresponding networks are HNNet^{MS6}, HNNet^{MS4}, and HNNet^{MS3}, respectively. The architecture of the improved version is presented in Fig. 3. The tensor size is expressed in the form of height × width × number of channels in the rest of the paper. The channel size will not

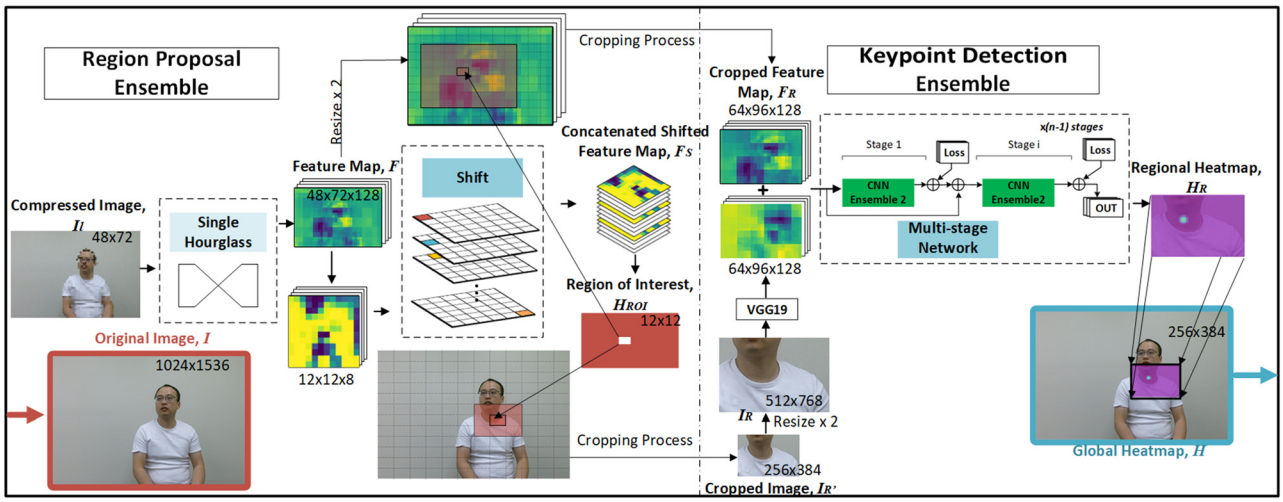

Fig. 3 Proposed hybrid neural network (HNNet) consisting of a region proposal ensemble (left) and a key point detection ensemble (right)

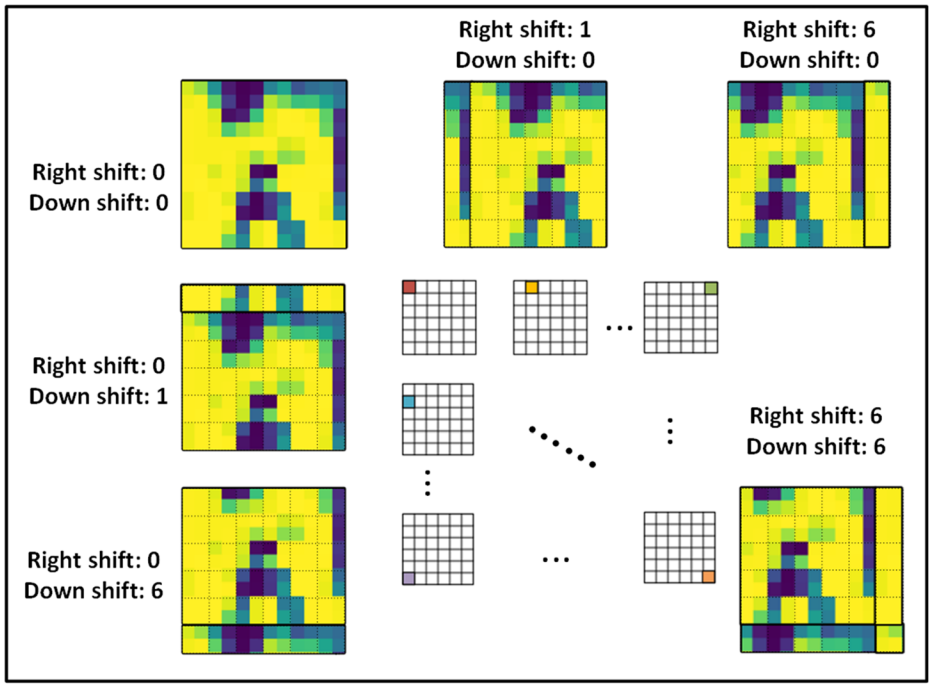

Fig. 4 Shift layer operation

be shown if it is equal to 1. These numbers are determined by hyper-parameters tuning during the training process.

2.1.1 Region Proposal Ensemble. The role of the region proposal ensemble is to provide the ROI and the useful spatial relationships of the image, I with a low cost. For this purpose, to feed the compressed version, I_l, of the original images with a size of $48 \times 72 \times 3$ into the ensemble is sufficient. This ensemble consists of a single hourglass model followed by a sequence of the shift [18] operations performed by rolling the resulting features in the horizontal or vertical direction, as shown in Fig. 4. It would eventually generate a one-hot map, H_{ROI}, of which the high-bit indicates the ROI, out of a predefined 12×12 grid. A feature map, F, of size $48 \times 72 \times 128$ is extracted from this ensemble.

The origin image I is cropped around the ROI location. The feature map F is also cropped around it with a larger span to preserve the necessary spatial information. The cropped image, I_R, of size 256×384, and the cropped feature map, F_R, of size $32 \times 48 \times 128$ are fed into the next ensemble for succeeding operations.

2.1.2 Keypoint Detection Ensemble. For the keypoint detection ensemble of HNNet^{MS6}, HNNet^{MS4}, and HNNet^{MS3}, the resulting cropped image, I_R, from the region proposal ensemble is up-sampled to the size of $512 \times 768 \times 3$ while the cropped features, F_R, is up-sampled to the size of $64 \times 96 \times 128$. The keypoint detection ensemble takes in both I_R and F_R, but at different layers of the neural network to make the final prediction. I_R is concatenated with F_R after passing through the VGG-19 model. These concatenated tensors then pass through a convolutional layer and the remaining multi-stage modules to generate the heatmap, H_R, of size 64×96. The high bit in H_R represents the predicted location of the CTM in the local coordinates. The regional heatmap is then padded to the global coordinates and generates the global heatmap, H, of size 256×384.

For HNNet^{MS6}, HNNet^{MS4}, and HNNet^{MS3}, the training processes of the region proposal ensemble and the keypoint detection ensemble are separate. First, the region proposal ensemble is trained. The optimized parameters of the region proposal ensemble are stored. The well-trained region proposal model will be implemented in the data generation process. The cropped image, I_R, and the cropped feature map, F_R are generated as the inputs to the keypoint detection model with the region proposal model, which predicts the ROI and provides the feature map F. In the validation process of the proposed networks, the two ensembles are combined as an end-to-end model. The model takes in both I and I_l as inputs and generates H as the output.

2.2 Manipulator Control. After the position of the CTM is detected, a robotic manipulator controller needs to plan global trajectories for the end-effector of the robotic manipulator to reach the CTM area and perform the Cricothyrotomy. As an initial step toward this goal, the manipulator control part of this paper focuses on guiding a robotic manipulator to the CTM. The neural network-based controller is trained in simulated environments powered by MuJoCo physics engine [19] using Reinforcement Learning. Binary sparse reward [20], instead of complex shaping reward, is used in the simulated environments to reduce the reward function design burden [21]. In this section, the detailed design of the neural network and simulated environment are presented.

2.2.1 Control Agent Neural Network. To control the robotic manipulator to perform the reaching task, an Actor-Critic method, deep deterministic policy gradient (DDPG) [22], was used in the training process to obtain the desired control policy. In this sparse reward environment, this offline approach was further improved using a policy learning method, hindsight experience replay (HER) [21]. To obtain a good generalization over the entire workspace of the robot, hyperparameter tuning is performed to obtain a dedicated parameter set for the neural network of both the actor and the critic. Table 1 presents the hyperparameters used for the tuning process. In this work, both the deterministic policy and value function are represented as two hidden-layer multilayer perceptrons (MLPs) with rectified linear unit (ReLU) activation functions. The output of the deterministic policy was further bounded in consideration of the actuator limits of the robotic manipulator.

2.2.2 Simulated Environments. To train the control agent, a 6-DOF Kinova JACO arm with a three-finger gripper as the end-effector is used in the simulation environments. The workspace of the simulated environment was set to fit the MICO arm, the actual robot arm that was used for the experiments. In this reaching task, the finger actuators of the gripper are fixed without control inputs, and the gripper orientation is fixed toward the floor at all times. In the reaching task, the arm needs to reach a randomly generated desired location above the floor, starting from a random gesture. The time-step of the simulation was set to 0.002 s to perform a fast and accurate simulation of the dynamic model.

Observations: To provide a more generalizable method that can be applied to different types of robotic manipulators, the states of the system described in the MuJoCo engine consist of the robotic gripper position and velocity in the workspace (the robotic manipulator world coordinates).

Actions: Instead of controlling the joint angles of the robot directly, the relative movement of the robotic gripper in the workspace was used as the control input for the robotic manipulator.

Table 1 Hyperparameter tuning for critic and actor

Parameters	Critic	Actor
Hidden layers #	[2, 3, 4]	[2, 3, 4]
Neurons #	[128, 256, 512]	[128, 256, 512]

Table 2 Summary of the dataset collection process

Diversity statistics of the subjects

Race		Age		Gender		Weight	
Mongoloid	6	18–21	2	Male	10	100 lb–130 lb	3
Caucasian	2	22–25	7			130 lb–160 lb	5
Black	5	26–29	4	Female	3	160 lb–200 lb	5

This allows for transferability among different robotic manipulators. The action output from the deterministic policy can be expressed as $A = \{a_i : a_i \in \mathbb{R}^3\}$.

Goals: The goals are defined as the target positions in the workspace that the gripper of the robotic manipulator is supposed to reach within a fixed period, $G = \{g_i : g_i \in \mathbb{R}^3\}$. In each training episode, both the goal and the initial position of the robotic gripper are randomly generated within a feasible workspace.

Rewards: In both teacher and student training environments, sparse rewards are used as, $r_t(s_{t+1}, g) = -(||f_g(s_{t+1}) - g|| > \varepsilon)$, where f_g maps the state, s, to an achieved goal, g, and ε determines the control precision in the task.

3 Training and Experiment

3.1 Cricothyroid Membrane Detection

3.1.1 Cricothyroid Membrane Dataset. A dataset containing 16,415 images was created with the visibility and pixel location (if visible) of the cricothyroid membrane on each image to train and test the proposed cricothyroid membrane keypoint detection neural network. The dataset contains images from 13 subjects with different genders, races, ages, and body shapes. The statistics of the subjects are provided in Table 2.

RGB image data was collected with A Kinect V2 RGB-D camera [16] from the subjects, as shown in Fig. 5(*a*). During the data collection process, each subject was asked to move his/her neck in three ways: (1) rotate the neck from side to side, (2) extend the neck to lift the chin upward, (3) bend the neck laterally to bring the ear to the shoulder, as shown in Figs. 5(*b*)–5(*g*).

For each of the movements mentioned above, images were captured from different points of view. The points of view were determined by the combinations of the different relative heights of the camera to the subject, h, the relative horizontal distance, w, and the angle between the neutral axis of the camera and the one of the subject, ψ, as shown in Fig. 5.

In the process of image collection, the camera captured 50 images for each combination. In total, 58,500 images were captured. 16,415 images that provide unobstructed views of the human face and neck area were chosen to build the dataset.

To make the annotation process of the dataset efficient and accurate, we built a MATLAB-based Graphical user interface (GUI) labeling program, as shown in Figs. 6(*a*)–6(*c*). In this program, the visibility of the keypoint is selected from three different levels: '0' (invisible in the image), '1' (visible, with no distinct feature), and '2' (visible, with distinct features). The location of the keypoint (on the pixel) is labeled if it is visible. The labeling process can be done with a few mouse-and-keyboard operations, which are explained in Fig. 6(*d*). The labeling is done by a medical provider in Respiratory Department.

3.1.2 Training Process and Results. Among the 16,415 images in the dataset, 80% (13132 images) were randomly selected as the training set and the other 20% (3283 images) were selected as the test set, such that the model is trained with sufficient number of samples to learn the patterns and perform well during the test phase, and also has sufficient test samples to be fairly evaluated. The original RGB images collected from the Kinect V2 camera had a resolution of 1080×1920. To accommodate the size of the neural networks, they were cropped around the

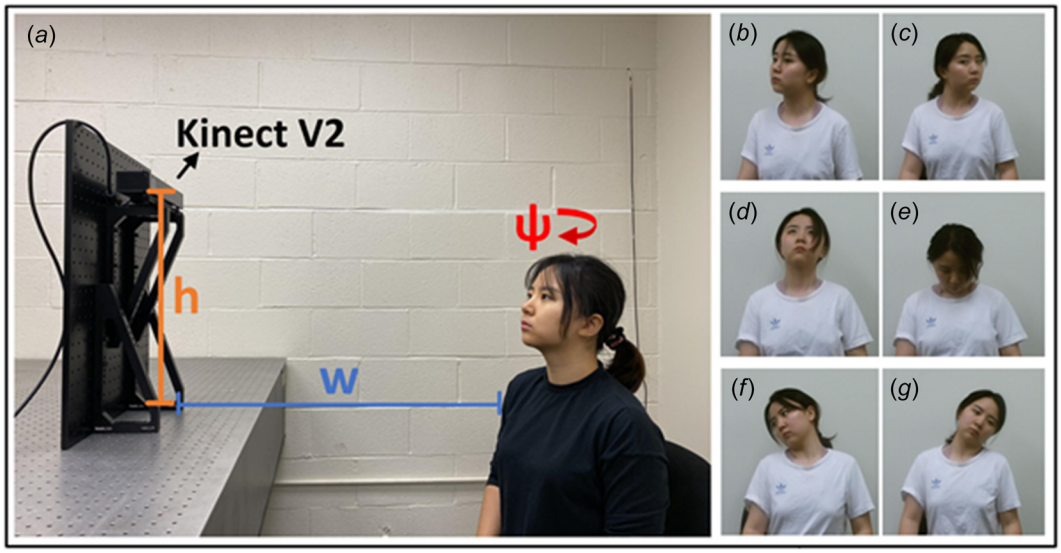

Fig. 5 (*a*) Image collection process, (*b*)–(*g*) demos of neck movements, (*b*) and (*c*) rotate the neck from side to side, (*d*) and (*e*) extend the neck to lift the chin upward, and (*f*) and (*g*) bend the neck laterally to bring the ear to the shoulder

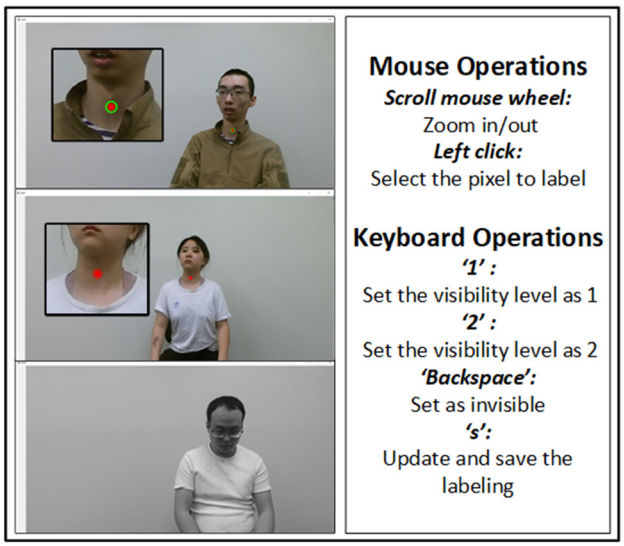

Fig. 6 (*a*)–(*c*) Screenshots of MATLAB GUI for the labeling process for cases when the visibility level of the keypoint is lableled as (*a*) '2', (*b*) '1', (*c*) '0', and (*d*) the instructions to label the image with the GUI

center to the size of 1024×1536. The augmentation process of the dataset consisted of rotation ($-30\,\text{deg} \sim 30\,\text{deg}$), scaling ($0.8 \sim 1.2$), and transportation ($0 \sim 1/2$ of the distance from the keypoint location to each edge). The images were also normalized to $0 \sim 1.0$ on all RGB channels.

In the paper, the proposed region proposal model from our previous work [14] was inherited. $P^{\text{PD}=0}$ and $P^{\text{PD}=1}$ of the region proposal network model are 73.1% and 99.7%, respectively, where $P^{\text{PD}=n}$ stands for the percentage of predictions with an error of less than n pixels in Euclidean distance. The average time taken for a single prediction on one image is 23.8 ms. The input and output sizes for this model are 48×72 and 12×12, respectively. The summary of the proposed region proposal model in comparison with other baseline models is provided in Table 3.

For the keypoint detection models, the groudtruth are heatmaps with 2-D Gaussian distribution centered on the keypoint location. Let $x_j \in \mathbb{R}^2$ be the keypoint location. The value, S_j, of each pixel, $x \in \mathbb{R}^2$, of the of heatmap is expressed as follows:

$$S_j = \exp\left(-\frac{||x - x_j||_2^2}{\sigma^2}\right) \tag{1}$$

where σ is a constant that controls the spread of the high bits. The neural networks were trained using Keras with Tensorflow as the backend on an NVIDIA Xp GPU. All models were trained for 20 epochs.

The keypoint detection models of HNNet^{MS6}, HNNet^{MS4}, and HNNet^{MS3}, with the number of convolutional-layers stages of 6, 4, and 3, were trained. A bare VGG19 multi-stage Network model was also trained for comparison. All the models were trained on the same dataset with the input size of $512 \times 736 \times 3$, and output size of 64×96. To accommodate the GPU memory and optimize the training process performance, the batch size was set to 8 for the keypoint detection models. A Euclidean loss was chosen as the loss function

$$\text{Eucl} - \text{loss} = \left(\frac{1}{2}\right) \sum_{i=1}^{n} \left(y_{\text{true}} - y_{\text{predict}}\right)^2 \tag{2}$$

Multi-SGD with the learning rate of 2×10^{-5} was chose as the optimizer to converge the models. The number of the trainable parameters of HNNet^{MS6}, HNNet^{MS4}, and HNNet^{MS3} are

Table 3 Summaries of training processes and testing result of region proposal model

Models	PRP[a]	Single hourglass	Stacked hourglass
Training process			
Batch size	32	32	24
Training parameter	3,505,827	3,426,163	6,562,470
Prediction accuracy (%)[b]			
$P^{\text{PD}=0}$[c]	73.1	69.5	70.0
$P^{\text{PD}=1}$	99.7	99.4	99.6
Running time (ms)[d]	23.4	12.9	21.0

[a]PRP stands for proposed region proposal model
[b]Based on results of prediction of 1946 images from testing dataset with CTM labeled as visible.
[c]$P^{\text{PD}=n}$ stands for the percentage of the images that the Euclidean distance between the predicted position of CTM and the ground truth is less than n pixels.
[d]The average time taken for a single prediction on one image (The models run on 1000 images in total).

Table 4 Summary of training processes and testing results of keypoint detection models

Models	HNNet^{MS3}	HNNet^{MS4}	HNNet^{MS6}	Multi-stage net	HNNet^{HG}
Training process					
Training parameter		20,028,484	28,103750	28,037,953	1,453,376
Prediction accuracy (%)[a]					
$P^{PD=5}$[b]	48.7	47.1	51.0	8.0	29.1
$P^{PD=10}$	85.0	86.5	87.2	32.5	61.7
$P^{PD=20}$	93.8	94.7	95.1	86.1	78.3
$P^{PD=30}$	95.8	96.6	96.7	96.8	84.3
Running time (ms)[c]	23.4 + 62.2	23.4 + 74.0	23.4 + 98.5	92.8	23.4 + 28.8

[a]Based on results of prediction of 1946 images from testing dataset with CTM labeled as visible.
[b]PD (pixel deviation) stands for Euclidean distances in pixels, and $P^{PD=n}$ stands for the percentage of predictions with an error of less than n pixels in Euclidean distance.
[c]The average time taken for a single prediction on one image (with 1000 prediction ran in total).

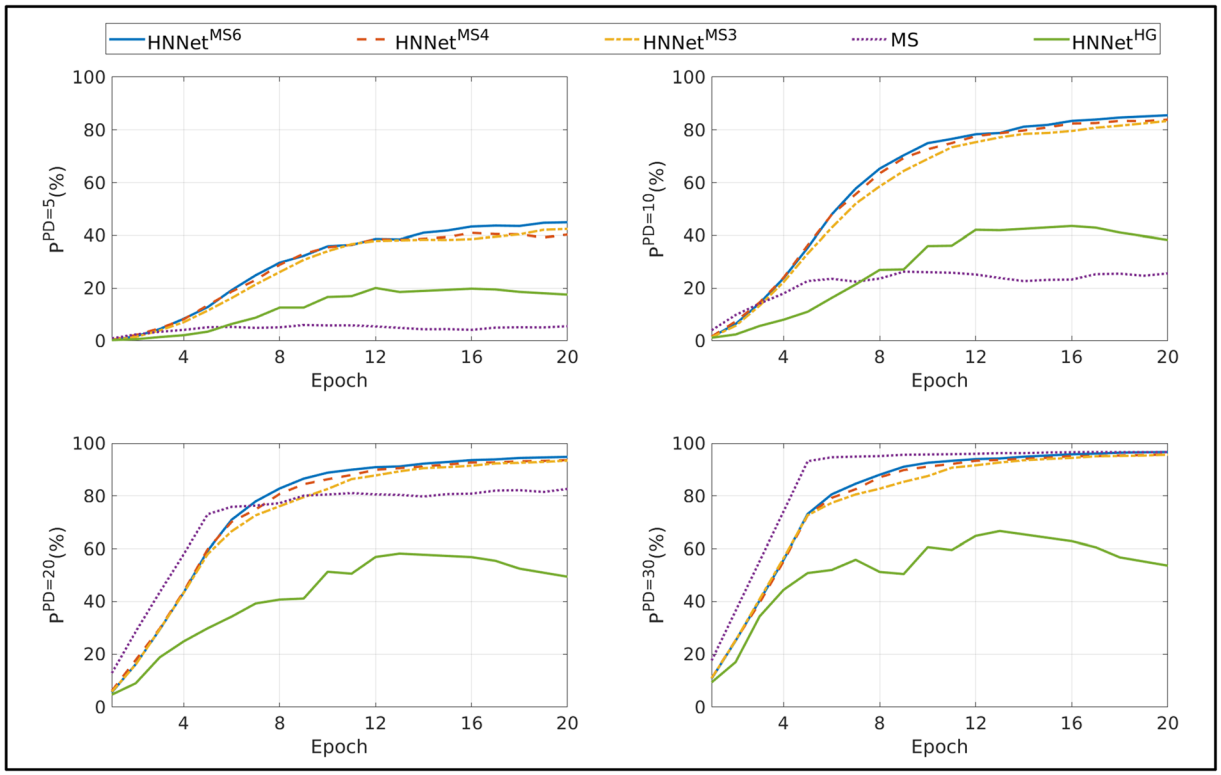

Fig. 7 Testing precision of keypoint detection models

28,103,750, 20,028,484, and 15,990,851, respectively. The training details of the models are presented in Table 4.

During the test stage, the Euclidean distance between the predicted position of the cricothyroid membrane and the labeled position was used to calculate the pixel deviation (PD)

$$\mathrm{PD} = \sqrt{\left(x - k \cdot \left(x_p - \frac{1}{2}\right)\right)^2 + \left(y - k \cdot \left(y_p - \frac{1}{2}\right)\right)^2} \quad (3)$$

where x, y is the labeled location of the keypoint on the original high-resolution image, and x_p, y_p is the predicted location scaled back to the original high-resolution image. k is the scale factor from the heatmap, H, to the original image, I. The evaluation results are shown in Table 4 and Fig. 7. $P^{PD=n}$ stand for the percentage of test samples with a prediction error of less than n pixels in Euclidean distance. $P^{PD=5}$ to $P^{PD=30}$ provide measures of precision among different thresholds.

The maximum threshold of the Euclidean distance was set to be 30 pixels as it approximately corresponds to 5 mm in real-world coordinates, and in the actual cricothyroid membrane position estimation process, it is considered as a correct estimation if the detected position is within 5 mm of the midline between the lower bound and the upper bound of the membrane [4].

The average running time of HNNet^{MS4} is 97.8 ms. It consists of two portions: 23.8 ms for the proposed region proposal ensemble, and 74.0 ms for the proposed keypoint prediction. Among all models, HNNet^{MS4} met the real-time requirement that the prediction time per frame should be less than 100 ms and achieved an optimized performance at the same time.

3.2 Manipulator Control. The control agent neural network introduced in Sec. 2.2.1 was trained to control the JACO arm for the reaching task in the simulated environment. The agent was trained for 120 epochs from scratch and eventually reached a success rate of 100%.

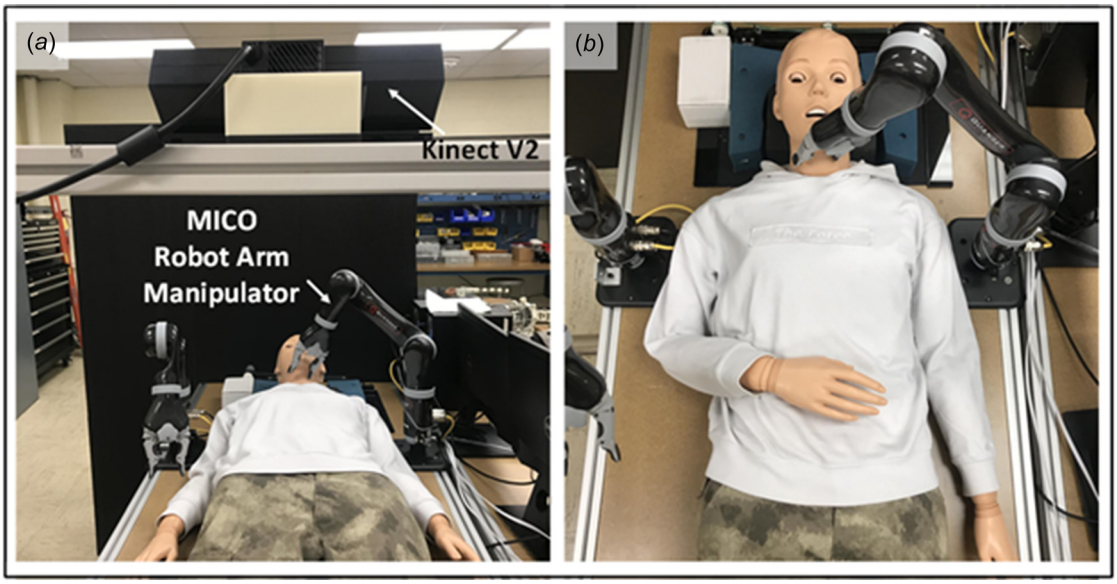

Fig. 8 Test bench setup (*a*) front view and (*b*) top view

Table 5 Coordinates of the devices on the test bench

Device i	Position[a] (m)	Orientation[b]
Robotic manipulator	$(0, 0.36, 0)$	I_3
Kinect v2	$(0.06, 0.03, 0.75)$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$

[a](x, y, z) in meter.
[b]With 1 as in same direction and -1 in opposite direction of (x, y, z).

Table 6 Computing hardware details of the integrated control system

Controller	Device	Function
High-level	AMD 1950X Nvidia 2080Ti	1. CTM detection 2. Manipulation planning in work space 3. Mapping planning to joint space
Low-level	Intel i7 6700	Manipulator joint position control

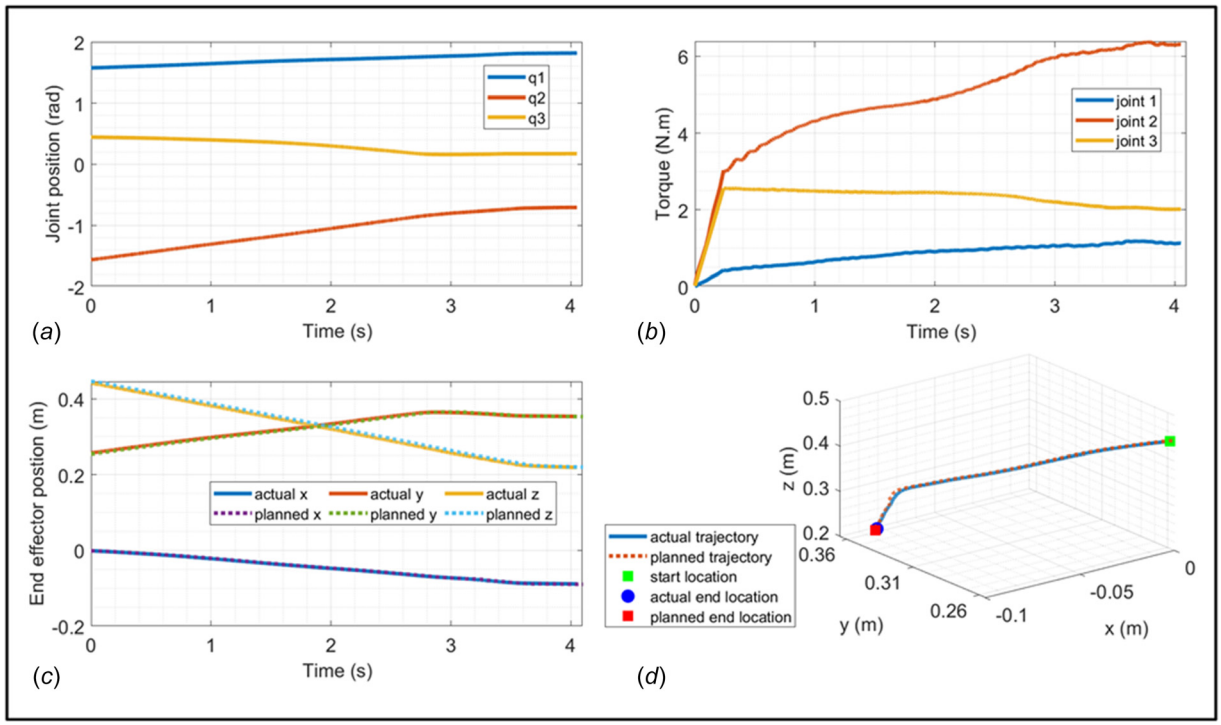

Fig. 9 The experiment result with the MICO robot arm manipulator (*a*) joint positions, (*b*) joint torques, (*c*) planned and actual end-effector positions, and (*d*) planned and actual trajectories

3.3 Experimental Validation

3.3.1 Experimental Setup. To experimentally validate the performance of the integrated CTM detection and manipulation system, a testbench was designed and setup to simulate the real-life situation inside the Robotics and Mechatronics Laboratory, as shown in Fig. 8. A Simple Simon manikin from Gaimard - a high-fidelity full-size manikin designed for medical training purposes - was used to simulate the patient. The manikin has sufficient details in its head-and-neck for our experiment. A Kinect V2 RGB-D camera was mounted on top of the manikin at the height of 0.75 m, and a MICO robot arm (with a control resolution of 5 mm) was placed on the right side of the manikin. The coordinates of the Kinect camera and the robot arm are calibrated with the hand-eye calibration [23] method. The summary of the coordinates of the devices is provided in Table 5. In this paper, the system was built based on the assumption that the victim was immobilized before and during the whole procedure.

The Kinect V2 was connected to the high-level controller with the CPU and the GPU in charge of the perception, decision-making, and control process. The RGB image of the upper body area of the manikin with size 424×512 captured from the Kinect V2 was sent to the computer, processed to the size of 1024×1536, normalized, and fed into the HNNet running on the GPU as input to generate the predicted CTM location in the x-y plane. The location of the point in the z-axis is then obtained from the depth image captured at the same time and the RGB/Depth-image mapping function provided by Kinect V2. The three-dimensional (3D) position is transferred to the robot arm coordinates. With the 3D position transferred to the robot arm coordinates as the desired position, the decision-making system with a well-trained neural network would generate the planned trajectory for the robot arm manipulator in the testbench coordinates. With the trajectory in the workspace, the inverse model controller generates the trajectory in the joint space, which is sent to the low-level controller to operate the robot arm manipulator. The details of the computing hardware are shown in Table 6.

3.4 Experimental Results. The planned trajectory of the MICO robot arm manipulator in the task space decided by the well-trained neural network and its actual trajectory are presented in Figs. 9(*c*) and 9(*d*), and the joint position and torque of the robot arm manipulator during the reaching process are presented in Figs. 9(*a*) and 9(*b*).

4 Conclusion

The paper focused on applying deep learning and reinforcement learning techniques to the tasks of CTM detection and robot arm manipulation. In this paper: (1) The HNNet model was proposed for precise real-time CTM detection; the model was trained and validated with the CTM dataset; (2) The robot arm was manipulated to reach the detected point using reinforcement learning; (3) The proposed techniques were combined into a single system, and the system was validated in real-life experiments on a human-sized medical manikin using a Kinect V2 camera and a MICO robot arm manipulator. Also, with the corresponding dataset provided, the HNNet can be applied to other human keypoint detection tasks with high-precision and real-time requirements. Ensemble-based architecture can also serve as a method to enhance the performance of the existing neural networks for various computer vision tasks.

The methods and results described in this paper serve as initial steps for the first-aid airway management robotic system to perform cricothyrotomy on a patient. Future work would be focused on the manipulation of the robot arm for operations in the next steps and improvement of the perception system. The first-aid airway management robotic system is designed to perform cricothyrotomy on the patient with a commercial cricothyrotomy kit. In this system, with the detected position of the CTM, the robot arm manipulator would pick up the cricothyrotomy needle and perform the incision. The CTM detection neural network will only provide the initial judgment of the location of the incision. However, it would require the topography of the surrounding region to be measured by precise instruments such as a force-sensitive resistor. With the 3D information, a more dependable estimation of the incision point can be achieved. The correct angle of the incision can also be determined from the information. Instead of using the analytical method, a neural network approach could be applied to solve the inverse kinematics of the robot arm to achieve a more robust performance when the task becomes increasingly complex. Moreover, the proposed architecture and methodologies described in this paper are not restricted to the application of airway management. A wide range of first-aid operations that requires high-precision human keypoint detection and robot arm manipulation, such as bleeding control, CPR, etc., could be developed by applying the framework of the proposed integrated system. The ideology of autonomous first-aid robotic systems with Artificial Intelligence could be more complete with gradual development and improvement in the future.

Furthermore, the system was built based on the assumption that the target was immobilized before and during the whole procedure. However, in real-life scenarios, the patient may experience movement in the head and neck caused by unstable support. The head support system of the SAVER will be implemented to stabilize the head and neck of the patient after an estimation of the CTM position is made with HNNet. Therefore, the robot arm manipulator will be able to operate on an immobilized target.

The comprehensiveness of the dataset can also be further enhanced by adding the elderly and people suffering from obesity to the pool of subjects, collecting the images from a varied range of backgrounds, and adding the simulated traumas to the collected images. With complicated factors such as wrinkles, fat, traumas, obstructive clothing, noisy backgrounds, etc., presented in the dataset, the detection neural network will be able to extract the features more accurately and achieve improved performance in the CTM detection under different unknown situations.

Acknowledgment

The authors would like to acknowledge the support of the medical expert, Zhiping Liu, in the Respiratory Department of XuZhou Central Hospital for labeling the position of the cricothyroid membrane in our dataset. The authors would also like to gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. This work was supported in part by the U.S. Army Medical Research & Material Command's Telemedicine & Advanced Technology Research Center (TATRC), under Contract No. W81XWH-16-C-0062. The views, opinions, and/or findings contained in this report are those of the authors and should not be construed as an official Department of the Army position, policy, or decision unless so designated by other documentation.

Funding Data

- U.S. Army Medical Research & Material Command's Telemedicine & Advanced Technology Research Center (TATRC) (Contract No. W81XWH-16-C-0062).

Data Availability Statement

The authors attest that all data for this study are included in the paper.

References

[1] Macdonald, J. C., and Tien, H. C., 2008, "Emergency Battlefield Cricothyrotomy," Can. Med. Assoc. J., **178**(9), pp. 1133–1135.

[2] Pracy, J., Brennan, L., Cook, T., Hartle, A., Marks, R., McGrath, B., Narula, A., and Patel, A., 2016, "Surgical Intervention During a Can't Intubate Can't Oxygenate (CICO) Event: Emergency Front-of-Neck Airway (FONA)?," Br. J. Anaesth., **117**(4), pp. 426–428.
[3] Meyer, T., and Patel, S., 2014, "Surgical Airway," Int. J. Crit. Illn. Injury Sci., **4**(1), p. 71.
[4] Aslani, A., Ng, S. C., Hurley, M., McCarthy, K. F., McNicholas, M., and McCaul, C. L., 2012, "Accuracy of Identification of the Cricothyroid Membrane in Female Subjects Using Palpation: An Observational Study," Anesth. Analg., **114**(5), pp. 987–992.
[5] Katos, M. G., and Goldenberg, D., 2007, "Emergency Cricothyrotomy," Oper. Tech. Otolaryngol. Head Neck Surg., **18**(2), pp. 110–114.
[6] Lind, M. M., Corridore, M., Sheehan, C., Moore-Clingenpeel, M., and Maa, T., 2018, "A Multidisciplinary Approach to a Pediatric Difficult Airway Simulation Course," Otolaryngol. Head Neck Surg., **159**(1), pp. 127–135.
[7] Williams, A., Sebastian, B., and Ben-Tzvi, P., 2019, "Review and Analysis of Search, Extraction, Evacuation, and Medical Field Treatment Robots," J. Intell. Rob. Syst. Theory Appl., **96**(3–4), pp. 401–418.
[8] Marescaux, J., Leroy, J., Gagner, M., Rubino, F., Mutter, D., Vix, M., Butner, S. E., and Smith, M. K., 2001, "Transatlantic Robot-Assisted Telesurgery," Nature, **413**(6854), pp. 379–380.
[9] Chapman, P. L., Cabrera, L. D., Varela-Mayer, C., Baker, M. M., Elnitsky, C., Figley, C., Thurman, R. M., Lin, C.-D., and Mayer, L. P., 2012, "Training, Deployment Preparation, and Combat Experiences of Deployed Health Care Personnel: Key Findings From Deployed U.S. Army Combat Medics Assigned to Line Units," Mil. Med., **177**(3), pp. 270–277.
[10] Cardenas, A., Quiroz, O., Hernández, R., Medellin-Castillo, H. I., González, A., Maya, M., and Piovesan, D., 2021, "Vision-Based Control of a Mobile Manipulator With an Adaptable-Passive Suspension for Unstructured Environments," ASME J. Mech. Rob., **13**(5), p. 050908.
[11] Johnson, B. V., Gong, Z., Cole, B. A., and Cappelleri, D. J., 2019, "Design of Compliant Three-Dimensional Printed Surgical End-Effectors for Robotic Lumbar Discectomy," ASME J. Mech. Rob., **11**(2), p. 020914.
[12] Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y., 2017, "Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, July 22–25, Vol. 2017-January, pp. 1302–1310.
[13] Ren, S., He, K., Girshick, R., and Sun, J., 2017, "Faster R-CNN: Towards Real-Time Object Detection With Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., **39**(6), pp. 1137–1149.
[14] Han, X., Ren, H., and Ben-Tzvi, P., 2020, "Autonomous Cricothyroid Membrane Detection Using Neural Networks for First-Aid Surgical Airway Management," International Design Engineering Technical Conferences, Online, Aug. 17–19.
[15] Ren, T., Dong, Y., Wu, D., and Chen, K., 2018, "Learning-Based Variable Compliance Control for Robotic Assembly," ASME J. Mech. Rob., **10**(6), p. 061008.
[16] Zhang, Z., 2012, "Microsoft Kinect Sensor and Its Effect," IEEE Multimed., **19**(2), pp. 4–10.
[17] Simonyan, K., and Zisserman, A., 2015, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings, San Diego, CA, May 7–9, p. 1409.1556.
[18] Choi, P. J., Oskouian, R. J., Tubbs, R. S., Xu, S., Perez, M., Yang, K., Perrenot, C., Felblinger, J., Hubert, J., Butner, S., Ghodoussi, M., Wu, B., Wan, A., Yue, X., Jin, P., Zhao, S., Golmant, N., Gholaminejad, A., Gonzalez, J., and Keutzer, K., 2018, "Transforming a Surgical Robot for Human Telesurgery," Surg. Endosc., **28**(5), pp. 69–78.
[19] Todorov, E., Erez, T., and Tassa, Y., 2012, "MuJoCo: A Physics Engine for Model-Based Control," 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Algarve, Portugal, Oct. 7–12, pp. 5026–5033.
[20] Ren, H., and Ben-Tzvi, P., 2020, "Advising Reinforcement Learning Toward Scaling Agents in Continuous Control Environments With Sparse Rewards," Eng. Appl. Artif. Intell., **90**, p. 103515.
[21] Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., and Zaremba, W., 2017, "Hindsight Experience Replay," 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, Dec. 4–9, Vol. 30, p. 1707.01495.
[22] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D., 2016, "Continuous Control With Deep Reinforcement Learning," 4th International Conference on Learning Representations, ICLR 2016, Conference Track Proceedings, San Juan, Puerto Rico, May 2–4, p. 1509.02971.
[23] Li, W., Dong, M., Lu, N., Lou, X., and Sun, P., 2018, "Simultaneous Robot–World and Hand–Eye Calibration Without a Calibration Object," Sensors (Basel, Switzerland), **18**(11), p. 3949.