Towards a Feasible Hand Gesture Recognition System as Sterile Non-contact Interface in the Operating Room with 3D Convolutional Neural Network

Roy Amante A. Salvador and Prospero C. Naval, Jr.
E-mail: rasalvador1@up.edu.ph, pcnaval@up.edu.ph
Computer Vision and Machine Intelligence Group, Department of Computer Science
College of Engineering, University of the Philippines, Diliman, Quezon City, Philippines

Keywords: hand gesture recognition system, computer vision, 3d convolutional neural network, operating room

Received: February 16, 2021

Operating surgeons are constrained when interacting with computer systems as they traditionally utilize hand-held devices such as keyboard and mouse. Studies have previously proposed and shown the use of hand gestures is an efficient, touchless way of interfacing with such systems to maintain a sterile field. In this paper, we propose a Deep Computer Vision-based Hand Gesture Recognition framework to facilitate the interaction. We trained a 3D Convolutional Neural Network with a very large scale dataset to classify hand gestures robustly. This network became the core component of a prototype application requiring intraoperative navigation of medical images of a patient. Usability evaluation with surgeons demonstrates the application would work and a hand gesture lexicon that is germane to Medical Image Navigation was defined. By completing one cycle of usability engineering, we prove the feasibility of using the proposed framework inside the Operating Room.

Povzetek: Prispevek skuša dokazati izvedljivost uporabe globokega računalniškega sistema za prepoznavanje kretenj z roko v operacijski sobi.

1 Introduction

Hand Gesture Recognition (HGR) Systems for interfacing is now more interesting and relevant than ever. The development and deployment of contactless technology could be part of community preparedness and response during disease outbreaks. With the ongoing Coronavirus Disease pandemic (COVID-19), it behooves us to observe guidelines such as frequent hand sanitation, social distancing, and the wearing of personal protective attire to reduce the spreading of pathogens and contamination. This aseptic setting is more strictly enforced in the Operating Room (OR) domain.

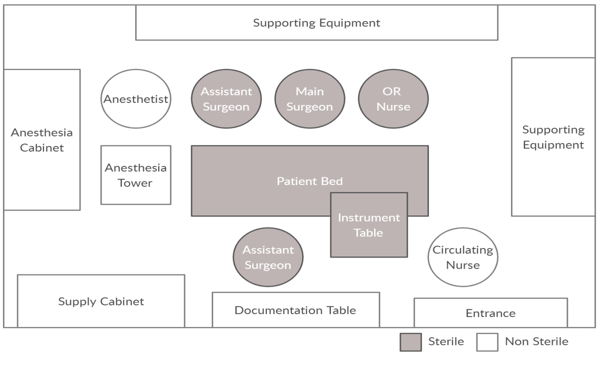


Figure 1: A typical operating room layout [1] depicting personnel and objects which should remain sterile during operations.

During operations, surgeons are not able to physically touch unsterile objects due to safety and health regulations but may need to control a medical system such as navigating to the patient's medical images, viewing for reference, and manipulating them on the screen. A simple hand gesture like swiping in the air would be more convenient for the surgeon and still a compliant way of interfacing with the system. Such a touchless system driven by modalities like hand gestures helps maintain sterility. Figure 1 shows a visualization of the Operating Room setting in terms of sterility. Wipfli et al. [2] compared and evaluated the gesture-controlled approach versus assistant-controlled with guided instructions from the surgeon when it comes to manipulating images in a surgery setting. The former received significantly higher ratings on efficiency and surgeon satisfaction.

Building a Hand Gesture Recognition System is nontrivial. It may be challenging to develop a quick and reliable method of detecting and recognizing the human hand in dynamic environments such as the operating room. For example, the hand may vary in size, color, illumination, position, and orientation to the camera. Many local and global invariant features have been manually designed and engineered to cope with these variabilities such as Histogram of Gradients (HOG) [3], Countour Description [4], Hu Invariant Moments [5], Fourier Descriptors [6], and Karhunen-Loeve Transform [7]. As with the no free lunch theorem, none of the features are better all the time from the others but there are factors and situations where they perform relatively better.

Deep Learning is a branch of machine learning based on a set of techniques that attempt to model high-level data abstractions using multiple processing layers composed of complex structures and transformations. It has been applied and proven useful in fields including but not limited to computer vision, natural language processing, speech recognition, and knowledge representation. As a subset of Representation learning [8], Deep Learning has the advantage of learning features automatically from data whereas rule-based and classical machine learning approaches need manual engineering and extracting of hand-designed features. It can also eliminate the need for pre-processing steps such as Hand Tracking and Region Segmentation making the processing pipeline simpler and straight-forward (raw data in; prediction out).

We approach the issue of maintaining sterility in the Operating Room by proposing the usage of a Real-time Computer Vision and Deep Learning-based Hand Gesture Recognition framework. The main contributions of our work are the following:

1. Demonstration of the feasibility of the proposed framework inside the Operating Room by designing and building a basic Medical Image Navigation prototype that is positively evaluated by surgeons.

2. Definition of a Hand Gesture Lexicon that is suitable for Medical Image Navigation application and consequently also appropriate to use inside the operating room.

3. A new Dynamic Hand Gesture dataset we've collected during the development of the prototype application. It is medium-sized containing more hand gesture samples than the Sheffield Kinect Gesture (SKIG) [9] dataset.

2 Hand gesture recognition systems in the operating room

Pioneering work in this arena heavily applied traditional computer vision techniques for performing image preprocessing, hand detection, and hand tracking and used finite-state machine for gesture classification [10, 11]. Some of them had poor usability and caused fatigue for the users [12]. A classical machine learning approach was taken by Achacon et al. [13]. Their system called *REALISM* included only a few gesture classes. They first performed hand detection with Haar-like features and cascade classifier then employed Principal Component Analysis and Euclidean Distance matching from the samples of the classes to perform classification.

Jacob et al. [14] defined a set of gestures for navigating medical images in consultation with veterinary surgeons. They used the 3D trajectory (3Dt) of the hand as the feature and Hidden Markov Model (HMM) as the classifier. The computation of the hand trajectory relied however on the Skeletal Tracking feature of Microsoft Kinect. They also used the skeletal information to compute head and torso orientation for determining intentional gestures. Several other HGR systems in the operating room [1, 15, 16, 17] have used and depended on the Microsoft Kinect device. Another popular device is the Leap Motion Controller (LMC) which uses proprietary drivers to process and format data into frames of objects like hands and fingers [18]. Park et al. [19] developed a message hooking program called *GestureHook* to convert gestures into mouse and keyboard functions to their medical system. [20] applied a rule-based classification based on the 3D hand movement (trajectory) using the points returned by the LMC device.

With their ability to project holograms that can be accessed interactively with hand gestures, mixed reality headsets are now making their way inside the operating room to support surgical procedures. A study by Galati et al. [21] found that they can increase the surgeon's productivity but highlighted that the battery autonomy and the weight of the device which can cause physical stress and discomfort are points for improvement. Furthermore, these devices cost significantly higher than the other capture devices.

3 Proposed framework

We propose a Hand Gesture Recognition System using Deep Computer Vision to act as an interface of the surgeon to a medical image navigation application. To do this successfully, we aimed at training a deep network and developing a framework which classifies static and dynamic gestures:

1. *With High Accuracy* - Classification performance must be invariant to the user and background. It must be at least comparable in performance with baseline systems.

2. *In Real-time* - The system must be able to complete the processing pipeline at most 250 milliseconds. The average reaction time for visual stimuli ranges from 250 to 350 milliseconds [22]. Taking longer than this range would introduce a noticeable lag visually.

3.1 Data capture

Microsoft Kinect was first explored for capturing the hand gestures as it provides depth modality. With depth information, one can easily determine which objects or pixels are in the foreground and the background. Moreover, the depth sensor of Kinect is an infrared camera so the effect of lighting conditions, user's skin color and clothing, and the background were assumed to have small to no impact on system performance. After the introduction of the first-generation Microsoft Kinect in 2010, there have been recognition systems developed and research using the device [23, 24, 25]. The simpler, cheaper and more available way is to use a single regular camera to capture the user's gesture. Many laptops have webcams and they are much widely available especially in developing countries. Usage of a single webcam also works in the proposed framework.

3.2 Network architecture

We determined the separation of networks for static hand postures and dynamic gestures is unnecessary as static hand gestures can also be seen as dynamic. Even though a static hand posture may not move in space, it always moves forward in time. With this, we chose a 3D Convolutional Neural Network as they are well-suited for systems involving spatiotemporal feature learning [26, 27, 28]. Specifically, it is a slightly modified version of the C3D model which was used for action recognition, action similarity labeling, and scene classification [29]. It resembles a VGG-11 [30] architecture replacing all 2D convolutional layers with 3D convolutional layers followed by Batch Normalization. The input to the network is the 16 RGB and depth fused frames, sized 96 x 96. The sizing of the input and network depended on the capacity of the GPU (NVIDIA GTX 970M graphics card) of the demo laptop machine used. Prediction of one sample took around 60 ms on battery and 40 ms when plugged in on the mentioned machine. We also verified the network with the SKIG [9] dataset using RGB only, and both RGB and depth (RGBD) on three-fold cross-validation. Results can be seen in Table 1. There might be other network architectures that are arguably better but our purpose is to only be able to predict robustly and quickly based on our defined guidelines for accuracy and real-time processing speed.

3.3 System actions / functionalities

The system actions include the ten functionalities listed by Jacob et al. [14]. The set was suggested by surgeons who were asked to give the most common functions they perform with medical images during surgeries. These are actions for navigation, brightness, orientation, and zoom level manipulation. Furthermore, we've added auxiliary functionalities for usability guided by consultation with a surgeon. These are locking/unlocking the system, panning, and animation of the images in the medical series. The Lock and Unlock actions signal to the system the user's intent to use the system. Locking the system reduces the possibility of recognizing unintended hand gestures performed by the user. The Panning action is seen as a supporting functionality when zoning into regions of interest. While the series animation functionalities not only enable viewing of the medical series but also help with navigating to a specified image as a series can contain hundreds or thousands of images/slices.

3.4 Medical image navigation interface

Developed in OpenCV [31], the interface is designed as a Medical Image/DICOM Viewer-like application. Digital Imaging and Communications in Medicine (DICOM) is the international standard format for storing medical imaging and information. In consultation with radiologists and surgeons, we were given and chose anonymized medical images taken from a real case with Appendicitis. We extracted and organized its images for the application to display as sample patient data.

3.5 Processing pipeline

As a real-time system, the network continuously gives its prediction for every new frame it receives from the camera but we need to treat the gestures of the user as discrete actions. We expect the user to perform each intentional gesture for about 1 second. For System Actions to be triggered, the prediction for the particular gesture class must be sustained by a set duration - gestures must continuously be predicted by the network in τ timesteps. If this condition is not met, we treat them as unintended gestures hence won't trigger any action. The workflow for executing System Actions can be summarized by the diagram in Figure 4. On our demo laptop with 16 GB memory, Intel Core i-7 processor, and NVIDIA GTX 970M graphics card, the application logic and rendering took around 50 ms running on battery and around 25 ms when running on power. Combining this with the network's prediction time, one cycle takes around 110 ms on battery, and 65 ms when running on power. This means the system's performance is in the range of about 8 to 15 fps and the suitable duration threshold τ is within this range.

4 Training our hand gesture recognition network

In this section, we discuss the efforts carried out in training the network used in our prototype application. Due to the size of the datasets, all network training efforts were performed in a Google Cloud Instance with 24 GB of memory and P100 GPU. We used Lasagne [32] deep learning library to train the networks using Stochastic Gradient Descent (SGD) with Nesterov Momentum with learning rate from $1e^{-2}$ to $1e^{-6}$ annealed by a factor of 0.1 whenever performance on the validation set did not improve. Moreover, we employed heavy Data Augmentation by manipulating the frames of the video samples - scaling inward and outward, random brightness and contrast, applying Gaussian noise, and random and multiple frame sampling of the training video clips.

4.1 Collecting and using our dataset

We first naively collected our dataset with the Microsoft Kinect camera. Some example RGB frames with their cor-

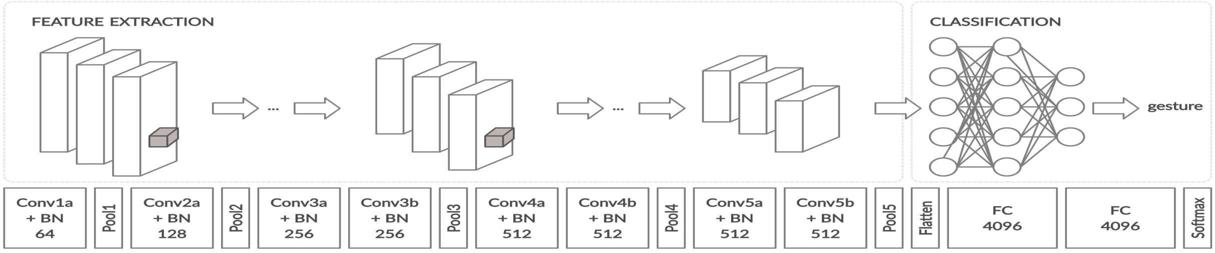


Figure 2: Modified C3D [29]. The network contains eight 3D convolutional layers (Conv*xx*) having a kernel size of 3 × 3 × 3 with a stride of 1 in all dimensions. The number of filters is indicated in their corresponding boxes. Each convolutional layer is followed by Batch Normalization (BN). There are five 3D max-pooling layers (Pool*x*) each with pooling kernel size and stride of 2 × 2 × 2, except for *Pool1* which is 1 × 2 × 2. The series of convolutional, batch normalization and pooling layers act as the feature extraction phase producing 3D feature maps. The network is closed off with the classification phase by two fully connected layers (FC) with 4096 neurons each and the output softmax layer whose size is the number of gesture classes.

Table 1: Accuracy of the Modified C3D Model on the SKIG Dataset. We first validated our chosen network architecture (Modified C3D Model) with the Sheffield Kinect Gesture (SKIG) [9] dataset. Data is divided into three folds each fold contains all samples from two subjects - Fold A (subjects 1 and 2), Fold B (subjects 3 and 4), and Fold C (subjects 5 and 6). For RGB only, and RGB with depth (RGBD) modality, the accuracy is more than our benchmark accuracy of around 93%.

	Fold A	Fold B	Fold C	Avg±Stdev
RGB	92.22%	94.17%	95.00%	93.8%±1.43%
RGBD	93.06%	93.61%	98.33%	95.0%±2.90%

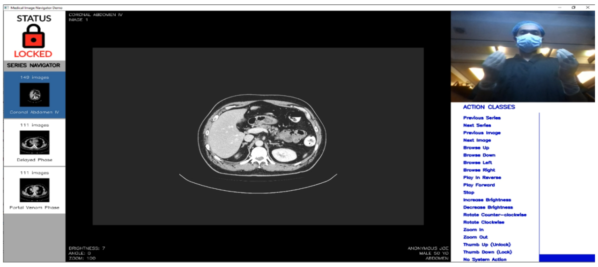


Figure 3: The Medical Image Navigation Application Interface. It is designed as a DICOM viewer-like application. The status (whether Locked or Unlocked) is displayed at the top-left corner. On the left-hand side is the list of available DICOM series for viewing. The center panel displays the current image/slice of interest of the current series. We can see the video stream and real-time visualization of the prediction of our system on the right-hand side of the application.

responding depth frames can be seen in Figure 5. Samples from four users were incrementally used for training and samples from the remaining user were used for validation. Adding more users to the training set was seen to increase the classification performance, however, at four users, the quality of the network is still very poor. The collection of new data and annotation is a long and tedious process so it was decided to look for a public dataset that can help boost performance via Transfer Learning.

4.2 Leveraging very large scale hand gesture dataset

The 20BN-JESTER [33] dataset is a very large scale hand gesture recognition dataset containing more than a hundred thousand densely-labeled clips performed by a huge number of crowd workers in front of a webcam or laptop camera. With transfer learning in mind, the selected network architecture is trained with the Jester dataset applying all previously mentioned regularization techniques. Since we have 4 channels as input to our chosen network and the Jester dataset only has RGB, a blank white image is fused in place of the depth frame. This is because, in Microsoft Kinect's depth images, white represents background. Samples in the dataset have varying lengths so, during prediction time, 16 frames sampled at equal intervals are extracted to represent the entire clip. Our trained network scored an overall accuracy of 94.52% on the official Jester validation set and around this value for the test set.

4.3 Fine-tuning

We initialized the network with the weights of our network trained with Jester Dataset and performed finetuning with our dataset. A sharp increase in performance as opposed to training from scratch is seen; however, the resulting System Accuracy is still quite poor at 67.58%. This might be because in the Jester dataset the gestures are performed directly in front of the camera so the hand is a dominant part

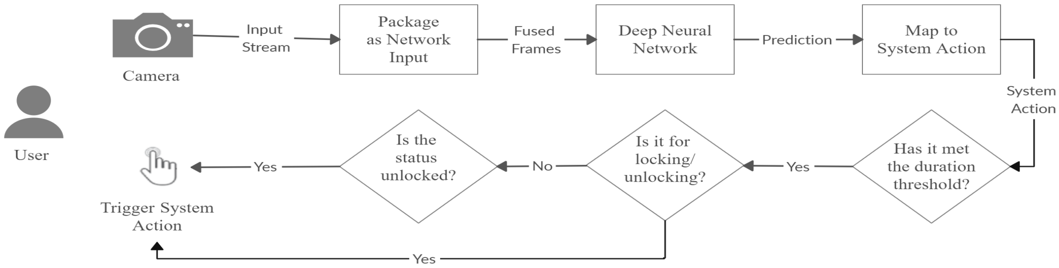


Figure 4: Overview of the Processing Pipeline. The camera continuously sends the video frames and the system collects the last 16 frames, resizes, and organizes them as input to our deep neural network. Every time the network performs an inference, the predicted gesture class is mapped to a system action. We check if the current system action has been sustained long enough for an actual gesture to take place. We further reduce the occurrence of triggering system actions for unintended gestures from the user by checking the application status (whether Locked or Unlocked).

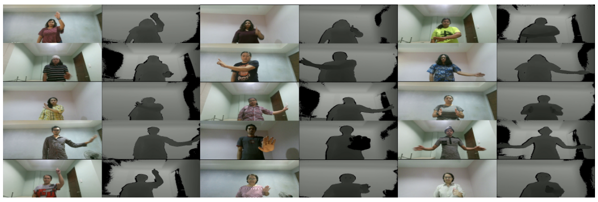


Figure 5: Our Dynamic Hand Gesture Dataset. Microsoft Kinect is used to capture the gestures of 5 users in RGB and depth modality. There are 16 gestures classes, 10 of which were taken from Jacob et al. [14], and a no system action class. Sequences are recorded with 3 varying backgrounds (blue, green and pink) and 3 different scales (user distance from the camera - 1, 2, 3 steps from the camera) performed each on their left and right hand unless the gesture needs two hands to perform. For each take, the users were asked to dress differently and apply different hairdos to increase variability. The dataset includes 1400+ gesture samples.

of the frame. In our dataset, gestures are performed standing with some distance from the Microsoft Kinect camera, making the hand relatively small compared to everything else in the frame. Another factor could be the Jester dataset only has the RGB modality. If it also has depth information, that might have further helped with boosting performance.

4.4 Utilizing Jester-trained network

At this point, we have trained a robust network capable of predicting for different users on different types of complex backgrounds. We assumed it would also work for a surgeon inside the Operating Room. Instead of forcing the use of our gesture lexicon, we used the ones in the Jester dataset and mapped the most intuitively matching gestures into our System Actions. Table 2 and Figure 6 details the mapping and its performance respectively. Table 3 summarizes the performance of training with the SKIG dataset, our dataset, and the Jester dataset. Compared to working with our dataset, utilizing the Jester-trained network meets the criterion for robust gesture classification.

5 Evaluation

5.1 Baseline systems

We evaluated against prior work on HGR systems for OR usage. The baseline systems include: *REALISM* [13], 3Dt+HMM without and with contextual cues (WC) [14], *GestureHook* [19], binarized (b-) and raw depth+2D-CNN [34], 3Dt+rule-based (RB) classification [20], and IR+ CapsNet [35]. They are listed in Table 4. We note that we compare the systems as a whole in which the number of action classes, types of gesture, and capture device are packaged and designed as a system. We addressed the collective weaknesses of these systems which include the need for manual engineering of features or pre-processing procedures to achieve a feasible recognition performance [13, 14, 34, 20], constraints on the hand gesture lexicon containing only static or movement hand gestures only but unable to process both due to methodology [13, 14, 34, 35], and reliance on the capabilities of the data capture device which are more expensive and may not be readily available in developing countries as an ordinary webcam would [14, 19, 34, 20, 35].

5.2 Test setup in the operating room

We were given a brief timeframe and have attempted to validate our system by testing it in an actual operating room. We recorded and annotated a sequence of gestures to see how it performs in real-time. Figure 7 shows the confidence over time of our trained network which is the output of the final softmax layer. Visually, the parts where there is sustained prediction confidence at 1.0 coincide with the ground truth. Quantitatively, the continuous recognition results in System Action Accuracy (SAA) of 94.95%, No Action Precision (NAP) of 96.54%, and No Action Recall (NAR) of 63.90% for the test sequence. The low NAR is attributed to confidence spikes for other gestures that corre-

Table 2: Jester Mapped Actions. Each gesture in the Jester dataset is intuitively mapped with our system actions.

Our System Action	**Jester Gesture**
Previous Series	Swiping Up
Next Series	Swiping Down
Previous Image	Swiping Left
Next Image	Swiping Right
Browse Up	Sliding Two Fingers Up
Browse Down	Sliding Two Fingers Down
Browse Left	Sliding Two Fingers Left
Browse Right	Sliding Two Fingers Right
Play Series In Reverse	Rolling Hand Backward
Play Series	Rolling Hand Forward
Stop Playing Series	Stop Sign Pushing Hand Away
Increase Brightness	Pushing Two Fingers Away
Decrease Brightness	Pulling Two Fingers In
Rotate Counter-clockwise	Turning Hand Counterclockwise
Rotate Clockwise	Turning Hand Clockwise
Zoom In	Zooming In With Full Hand Zooming In With Two Fingers
Zoom Out	Zooming Out With Full Hand Zooming Out With Two Fingers
Thumbs Up (Unlock)	Thumb Up
Thumbs Down (Lock)	Thumb Down
No System Action	No gesture Doing other things Pulling Hand In Drumming Fingers Shaking Hand

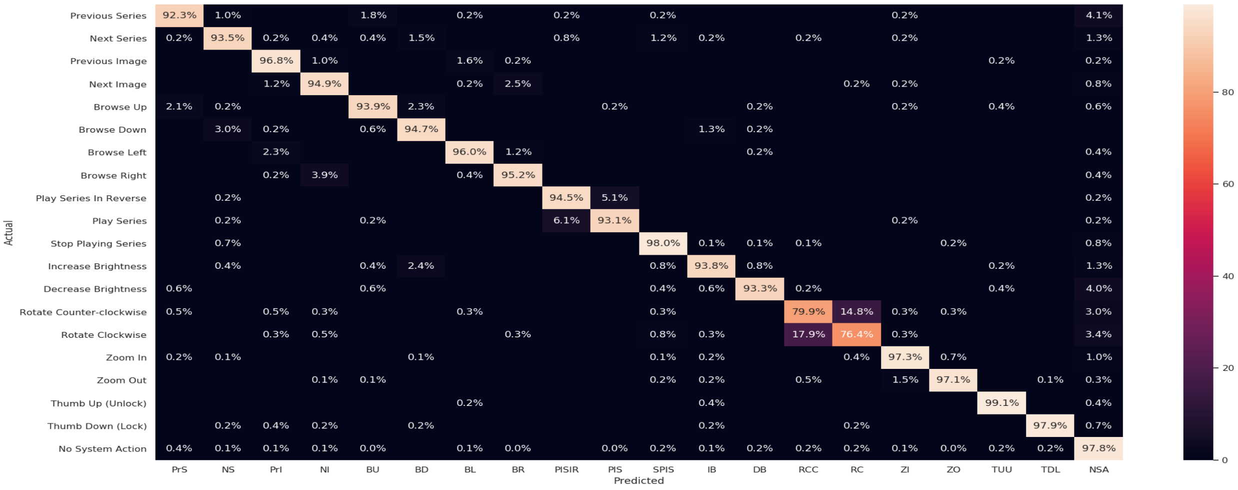


Figure 6: Jester Mapped Actions Performance. The confusion matrix of the proposed mapping.

spond to unintended gestures. They generally do not trigger system actions since we employed a set duration threshold to mitigate.

5.3 Usability evaluation with surgeons

We created an online survey form for the evaluation of the hand gestures as well as the overall usability of the application. There were 11 survey respondents from 7 in-

Table 3: Hand Gesture Recognition Results. We quantified the robustness of the trained network by the following: System Action Accuracy (SAA) tells us how good the network is in identifying system action classes, No Action Precision (NAP) shows how resilient the network is against classifying intended gestures as unintended (classified as no system action), and No Action Recall (NAR) denotes how resilient the network is against classifying unintended gestures as intended.

	SAA	**NAP**	**NAR**
SKIG Dataset (RGB)	93.80%	-	-
SKIG Dataset (RGBD)	95.00%	-	-
Own Dataset (1 user)	0.96%	79.18%	97.37%
Own Dataset (2 users)	23.58%	83.59%	75.38%
Own Dataset (3 users)	29.87%	86.15%	82.05%
Own Dataset (4 users)	43.57%	90.80%	88.93%
Jester Validation Set	94.52%	-	-
Jester Test Set	94.26%	-	-
Jester pre-training + Own Dataset (4 users)	67.58%	94.65%	91.44%
Jester Mapped Actions	94.48%	96.62%	97.80%

Table 4: Comparison with Baseline Systems. We relate the performance of our work with their reported System Action Accuracy (SAA) along with the type of gestures used (whether static, dynamic, or both), and the number of System Actions/commands.

	Capture Device	**Gesture Type**	**Number of System Actions**	**System Action Accuracy (SAA)**
REALISM [13]	webcam	static	5	$<$ 80%
3Dt+HMM [14]	Kinect	dynamic	10	93.60%
3Dt+HMM WC [14]	Kinect	dynamic	10	92.58%
GestureHook [19]	LMC	dynamic	8	$\approx$ 92%
depth+2D-CNN [34]	ToF	static	10	94.86%
b-depth+2D-CNN [34]	ToF	static	10	92.07%
3Dt+RB [20]	LMC	dynamic	10	95.83%
IR+CapsNet [35]	LMC	static	5	86.46%
Ours	webcam	both	19	94.48%

stitutions with varying experience ranging from fellows in post-residency training to attending physicians from different specialties such as General Surgery, Pediatric Surgery, Surgical Oncology, and Surgical Endoscopy. To avoid biased feedback, our main consulting surgeon did not participate. Participants were first introduced with the purpose of the research and then walked through the application by showing them images of the setup and video clip recordings of each system action being triggered in the system by their corresponding hand gesture. We asked them to choose from Strongly Disagree, Disagree, Neutral, Agree, and Strongly Agree for each criterion and we converted their answers numerically from 1 to 5 respectively for analysis. At the end of the form, they were also asked to evaluate and provide their overall thoughts on the usability of the system.

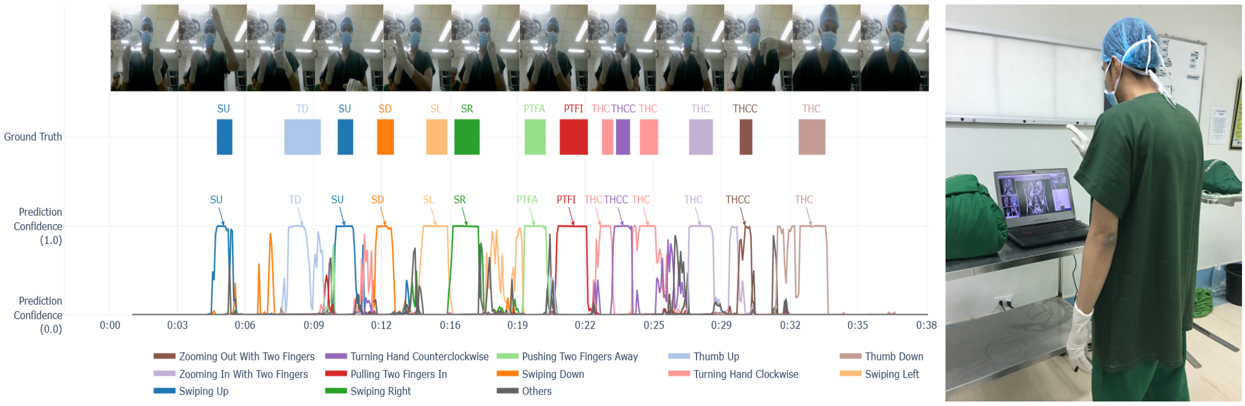
5.4 Hand gesture lexicon

For the hand gestures to be feasible, they should be positively rated on the following criteria [12]: Intuitiveness (Gesture Intuitively Matches the System Action Performed), Ease of Use (Gesture Is Easy and Comfortable to Perform), and Memorability/Ease of Remembrance. Additionally, we also included Appropriateness to use inside the OR. Table 5 shows the detailed ratings of the surgeons in our evaluation survey. All of the hand gestures received a mean rating greater than 3 (Neutral), and the majority of them greater than 4 (Agree). Most surgeons did not provide any suggestions to improve implying they were content with the gestures. With this, we can say that we have a workable initial set of hand gestures for the application just by using the ones in the Jester dataset.

A few of the hand gestures particularly those for brightness and orientation manipulation and playing the series received a relatively lower rating across all criteria. Some of the comments mention a preference for gestures involving smaller movements. Some suggested highly subjective gestures and additional functionalities with no agreement with other respondents. To determine the final set of proposed gestures for a Medical Image Navigation application in the operating room, we perform the following:

1. If the gesture received consistently lower scores across all criteria, we replace it with:

   (a) The suggested improved gesture provided by at



(a) (b)

Figure 7: Test inside the operating room. (a) Visualization of the continuous hand gesture recognition during the test. (b) The system hardware (a laptop with a webcam in this case) is placed in a convenient location adhering to sterility rules. To consult with the patient's medical images, the surgeon moves in front of the camera and gestures to the system.

least two surgeons in agreement in the evaluation; else with

(b) The suggested improved gesture provided by at least one surgeon in agreement with Jacob et al [14].

2. If there are no suggestions or if the gesture received satisfactory ratings, we keep the gesture.

The resulting improved hand gesture lexicon can be found in Table 6. We note that gestures can have different meanings across cultures hence some of them might only be suitable in our local setting.

5.5 Overall usability

For the overall usability of the application, we measure with the following criteria: Usefulness, Efficiency, Learnability, and Satisfaction. Figure 8 depicts the ratings given by the survey respondents. Feedback from surgeons is strongly positive with a mean score between 4 and 5 (from Agree to Strongly Agree) for all of our criteria. At 95% probability, the confidence intervals of the mean rating using t-distribution are 4.36 ± 0.62 for Usefulness, 4.09 ± 0.82 for Efficiency, 4.27 ± 0.56 for Learnability, and 4.36 ± 0.54 for Satisfaction. Only one surgeon disagreed with the usefulness of the application and the efficiency it would bring to productivity. The consensus is that the system is easily learnable and quite useful. The majority did not leave any further comments but some left suggestions on additional functionalities such as contrast management, measuring, incorporating voice commands, and using additional monitor screens to display the interface and video feed.

5.6 Limitations

Due to restrictions on physical meetings brought about by ongoing local policies regarding community quarantines, evaluating the system with multiple surgeons in an OR session has been a roadblock. However, we believe the results on the Jester evaluation set (a highly variable dataset which contains 14,000+ samples) coupled with our brief test in an operating room translates to a generally feasible hand gesture recognition performance.

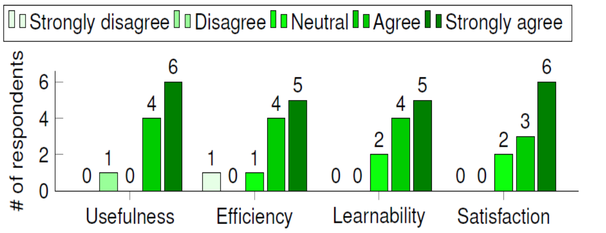


Figure 8: Evaluation survey overall usability rating distribution. Most of the respondents were generally satisfied, agreeing with the system's usefulness, efficiency, and learnability, suggesting high viability of the application.

For the framework to be seamlessly applied to any application in the operating room or any domain, there should be a capability for defining custom gestures to use [1]. With the current approach, it is difficult to achieve satisfactory performance for new gestures without acquiring a large enough number of samples as shown in Table 3. An extension of this work that could mitigate this issue is to integrate a One or Few-Shot Learning mechanism of hand gestures. [36] had executed this by using a pre-trained network for feature extraction then employing some distance measurement.

6 Conclusion

In this paper, we were able to demonstrate that a straightforward Deep Learning and Computer Vision-based framework is a viable solution in maintaining sterility in the op-

Table 5: Mapped Jester gestures evaluation results. Participants were asked to choose among strongly disagree (1), disagree (2), neutral (3), agree (4), and strongly agree (5) for each criterion for each gesture. The values displayed are the confidence intervals of the mean hand gesture ratings using t-distribution at 95% probability.

System Action	**Hand Gesture**	**Appropriateness**	**Intuitiveness**	**Ease of Use**	**Memorability**
Unlock	Thumb Up	4.00±0.74	3.54±0.70	4.09±0.36	4.27±0.31
Lock	Thumb Down	3.91±0.76	3.91±0.76	4.18±0.27	4.27±0.31
Previous Series	Swipe Up	4.09±0.36	4.09±0.36	3.91±0.56	4.09±0.36
Next Series	Swipe Down	4.00±0.42	4.00±0.42	3.91±0.47	4.00±0.42
Previous Image	Swipe Left	4.36±0.34	4.27±0.43	4.27±0.43	4.36±0.34
Next Image	Swipe Right	4.36±0.34	4.18±0.50	4.09±0.56	4.36±0.34
Play Series	Roll Hand Forward	3.73±0.68	3.73±0.61	3.45±0.35	3.64±0.69
Play Series In Reverse	Roll Hand Backward	3.91±0.56	3.64±0.62	3.45±0.76	3.64±0.69
Stop Series	Palm Facing Screen	4.45±0.35	4.18±0.72	4.36±0.45	4.27±0.53
Pan Up	Slide Two Fingers Up	4.18±0.27	3.91±0.47	4.18±0.27	4.09±0.36
Pan Down	Slide Two Fingers Down	4.18±0.27	4.00±0.42	4.09±0.36	4.09±0.36
Pan Left	Slide Two Fingers Left	4.27±0.31	4.18±0.41	4.27±0.31	4.27±0.85
Pan Right	Slide Two Fingers Right	4.27±0.31	4.18±0.41	4.27±0.31	4.27±0.31
Increase Brightness	Push Two Fingers Away	3.82±0.72	3.27±0.74	3.73±0.75	3.55±0.87
Decrease Brightness	Pull Two Fingers In	3.82±0.72	3.45±0.72	3.73±0.82	3.45±0.92
Rotate Counter-clockwise	Turn Hand Counter-clockwise	4.18±0.27	3.73±0.61	3.91±0.63	3.64±0.75
Rotate Clockwise	Turn Hand Clockwise	4.00±0.52	3.55±0.82	3.73±0.61	3.36±0.69
Zoom In	Open Two Fingers / Hand	4.45±0.35	4.45±0.35	4.36±0.45	4.45±0.35
Zoom Out	Close Two Fingers / Hand	4.45±0.35	4.45±0.35	4.36±0.45	4.45±0.35

erating room. We implemented an end-to-end, real-time, robust Hand Gesture Recognition System applied for usage inside the operating room in the form of a Medical Image Navigation application that is not dependent on the capture device and is positively evaluated by surgeons. General feedback from our local surgeons shows receptiveness and willingness to apply this technology. Furthermore, we defined a set of suitable hand gestures for the application. This set coupled with the framework can serve as a foundation for building and deploying Hand Gesture-controlled applications in our operating rooms as well as in other more lenient settings requiring sterility maintenance.

Acknowledgement

We would like to thank the following medical professionals who helped us with this work: Dr. Alvin Caballes from the University of the Philippines Philippine General Hospital (UP PGH), Department of Surgery, our primary surgeon in consultation with during the creation and evaluation of the prototype application; Dr. Cecilia Amicis and the Radiology Department of Veterans Memorial Medical Center (VMMC) who provided anonymized medical images from areal case study; and Dr. Jose Abellera and the Surgery Department at National Children's Hospital (NCH) who helped and allowed us to perform a brief test setup in one of their operating rooms.

Table 6: Suggested hand gesture lexicon for a medical image navigation application. An initial set of gestures was defined by mapping the System Actions with the Jester dataset [33] gestures. This was refined based on the results of our evaluation survey coupled with the ethnographic study conducted by Jacob et al [14]. It was raised that if there are added functionalities for confirmation, the thumb up and down gestures would be more appropriate for answering yes and no respectively.

System Action	**Hand Gesture**	**Details**
Unlock	Thumb Up	Thumb Up, other fingers tucked in.
Lock	Thumb Down	Thumb Down, other fingers tucked in.
Go To Previous Series	Swipe Up	Palm facing upward. Smaller movement of four fingers pivoting upward.
Go To Next Series	Swipe Down	Palm facing downward. Smaller movement of four fingers pivoting downward.
Go To Previous Image	Swipe Left	Palm facing camera or side. Move hand to the left.
Go To Next Image	Swipe Right	Palm facing camera or side. Move hand to the right.
Play Series	Swipe Down (Side View)	Side view of the hand facing screen. Palm facing downward. Smaller movement of four fingers pivoting downward.
Play Series In Reverse	Swipe Up (Side View)	Side view of the hand facing screen. Palm facing upward. Smaller movement of four fingers pivoting upward.
Stop Playing Series	Stop Sign	Open hand palm facing Screen.
Pan Up	Slide Two Fingers Up	Point index and middle finger. Move hand or two fingers upward.
Pan Down	Slide Two Fingers Down	Point index and middle finger. Move hand or two fingers downward.
Pan Left	Slide Two Fingers Left	Point index and middle finger. Move hand or two fingers towards the left.
Pan Right	Slide Two Fingers Right	Point index and middle finger. Move hand or two fingers towards the right.
Increase Brightness	Push Hand Away	Palm facing camera, move open hand towards camera. Taken from [14].
Decrease Brightness	Pull Hand In	Back of hand facing camera, move open hand away from camera. Taken from [14].
Rotate Counter-clockwise	Swipe Counter-clockwise	Palm facing camera, wave hand to the left (counter-clockwise). Taken from [14].
Rotate Clockwise	Swipe Clockwise	Palm facing camera, wave hand to the right (clockwise). Taken from [14].
Zoom In	Open Two Fingers	Index and thumb touching initially, other fingers tucked in. Move index and thumb away from each other.
Zoom Out	Close Two Fingers	Index and thumb away from each other initially, other fingers tucked in. Move index and thumb towards each other until they touch.

References

[1] A. Bigdelou, *Operating Room Specific Domain Model for Usability Evaluations and HCI Design*. PhD thesis, Technical University Munich, 2012.

[2] R. Wipfli, V. Dubois-Ferrière, S. Budry, P. Hoffmeyer, and C. Lovis, "Gesture-Controlled Image Management for Operating Room: A Randomized Crossover Study to Compare Interaction Using Gestures, Mouse, and Third Person Relaying," *PLoS ONE*, vol. 11, no. 4, p. e0153596, 2016.

https://doi.org/10.1371/journal.pone.0153596.

[3] W. T. Freeman and M. Roth, "Orientation histograms for hand gesture recognition," 1995.

[4] C.-C. Chang, I.-Y. Chen, and Y.-S. Huang, "Hand pose recognition using curvature scale space," vol. 2, pp. 386 – 389 vol.2, 02 2002. https://doi.org/10.1109/ICPR.2002.1048320.

[5] P. Premaratne, *Human Computer Interaction Using Hand Gestures*. Springer Science+Business Media Singapore, 2014. https://doi.org/10.1007/978-3-642-14831-6_51.

[6] S. Conseil, S. Bourennane, and L. Martin, "Comparison of fourier descriptors and hu moments for hand posture recognition," in *2007 15th European Signal Processing Conference*, pp. 1960–1964, 2007. https://doi.org/10.5281/zenodo.40606.

[7] J. Singha and K. Das, "Hand gesture recognition based on karhunen-loeve transform," *Mobile and Embedded Technology International Conference (MECON)*, 06 2013.

[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. https://www.deeplearningbook.org.

[9] L. Liu and L. Shao, "Learning discriminative representations from rgb-d video data," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pp. 1493–1500, AAAI Press, 2013.

[10] C. Grätzel, T. Fong, S. Grange, and C. Baur, "A non-contact mouse for surgeon-computer interaction," *Technology and health care : official journal of the European Society for Engineering and Medicine*, vol. 12, pp. 245–57, 02 2004. https://doi.org/10.3233/THC-2004-12304.

[11] J. Wachs, H. Stern, Y. Edan, M. Gillam, C. Feied, M. Smith, and J. Handler, "Gestix: A doctor-computer sterile gesture interface for dynamic environments," in *Soft Computing in Industrial Applications* (A. Saad, K. Dahal, M. Sarfraz, and R. Roy, eds.), (Berlin, Heidelberg), pp. 30–39, Springer Berlin Heidelberg, 2007. https://doi.org/10.1007/978-3-540-70706-6_3.

[12] A. Hurstel and D. Bechmann, "Approach for intuitive and touchless interaction in the operating room," *J*, vol. 2, pp. 50–64, 01 2019. https://doi.org/10.3390/j2010005.

[13] M. Achacon, David Louis Jr, D. M Carlos, M. Kaye Puyaoan, C. T Clarin, and P. Naval, "Realism: Real-time hand gesture interface for surgeons and medical experts," 09 2010.

[14] M. G. Jacob, J. P. Wachs, and R. A. Packer, "Hand-gesture-based sterile interface for the operating room using contextual cues for the navigation of radiological images," *J Am Med Inform Assoc*, vol. 20, pp. e183–186, Jun 2013. https://doi.org/10.1136/amiajnl-2012-001212.

[15] G. C. Ruppert, L. O. Reis, P. H. Amorim, T. F. de Moraes, and J. V. da Silva, "Touchless gesture user interface for interactive image visualization in urological surgery," *World J Urol*, vol. 30, pp. 687–691, Oct 2012. https://doi.org/10.1007/s00345-012-0879-0.

[16] M. Strickland, J. Tremaine, G. Brigley, and C. Law, "Using a depth-sensing infrared camera system to access and manipulate medical imaging from within the sterile operating field," *Can J Surg*, vol. 56, pp. 1–6, Jun 2013. https://doi.org/10.1503/cjs.035311.

[17] J. H. Tan, C. Chao, M. Zawaideh, A. C. Roberts, and T. B. Kinney, "Informatics in Radiology: developing a touchless user interface for intraoperative image control during interventional radiology procedures," *Radiographics*, vol. 33, no. 2, pp. 61–70, 2013. https://doi.org/10.1148/rg.332125101.

[18] B. Pavaloiu, "Leap motion technology in learning," pp. 1025–1031, 05 2017. https://doi.org/10.15405/epsbs.2017.05.02.126.

[19] B. J. Park, T. Jang, J. W. Choi, and N. Kim, "Gesture-Controlled Interface for Contactless Control of Various Computer Programs with a Hooking-Based Keyboard and Mouse-Mapping Technique in the Operating Room," *Computational and Mathematical Methods in Medicine*, vol. 2016, p. 5170379, 2016. https://doi.org/10.1155/2016/5170379.

[20] P. Sa-nguannarm, T. Charoenpong, C. Chianrabutra, and K. Kiatsoontorn, "A method of 3d hand movement recognition by a leap motion sensor for controlling medical image in an operating room," in *2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*, pp. 17–20, 2019. https://doi.org/10.1109/ICA-SYMP.2019.8645985.

[21] R. Galati, M. Simone, G. Barile, R. De Luca, C. Cartanese, and G. Grassi, “Experimental Setup Employed in the Operating Room Based on Virtual and Mixed Reality: Analysis of Pros and Cons in Open Abdomen Surgery,” *J Healthc Eng*, vol. 2020, p. 8851964, 2020. `https://doi.org/10.1155/2020/8851964`.

[22] J. Shelton and G. Kumar, “Comparison between auditory and visual simple reaction times,” *Neuroscience & Medicine*, vol. 1, pp. 30–32, 01 2010. `https://doi.org/10.4236/nm.2010.11004`.

[23] Y. Li, “Multi-scenario gesture recognition using kinect,” in *Proceedings of the 2012 17th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational & Serious Games (CGAMES)*, CGAMES ’12, (Washington, DC, USA), pp. 126–130, IEEE Computer Society, 2012. `https://doi.org/10.1109/CGames.2012.6314563`.

[24] A. Tang, K. Lu, Y. Wang, J. Huang, and H. Li, “A real-time hand posture recognition system using deep neural networks,” *ACM Trans. Intell. Syst. Technol.*, vol. 6, pp. 21:1–21:23, Mar. 2015. `https://doi.org/10.1145/2735952`.

[25] C. Yang, Y. Jang, J. Beh, D. Han, and H. Ko, “Gesture recognition using depth-based hand tracking for contactless controller application,” in *2012 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 297–298, Jan 2012. `https://doi.org/10.1109/ICCE.2012.6161876`.

[26] J. Li, S. Zhang, and T. Huang, “Multiscale 3d convolution network for video based person re-identification,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8618–8625, 07 2019. `https://doi.org/10.48550/arXiv.1811.07468`.

[27] D. Cheng, S. Xiang, C. Shang, Y. Zhang, F. Yang, and L. Zhang, “Spatio-temporal attention-based neural network for credit card fraud detection,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 362–369, 04 2020. `https://doi.org/10.1109/ICIP.2019.8803152`.

[28] P. Pandey, A. P. Prathosh, M. Kohli, and J. Pritchard, “Guided weak supervision for action recognition with scarce data to assess skills of children with autism,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 463–470, 04 2020. `https://doi.org/10.48550/arXiv.1911.04140`.

[29] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV ’15, (Washington, DC, USA), pp. 4489–4497, IEEE Computer Society, 2015. `https://doi.org/10.1109/ICCV.2015.510`.

[30] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014.

[31] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[32] S. Dieleman, J. Schlüter, C. Raffel, E. Olson, S. K. Sønderby, D. Nouri, *et al.*, “Lasagne: First release.,” Aug. 2015. `https://doi.org/10.5281/zenodo.27878`.

[33] J. Materzynska, G. Berger, I. Bax, and R. Memisevic, “The jester dataset: A large-scale video dataset of human gestures,” in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 2874–2882, 2019. `https://doi.org/10.1109/ICCVW.2019.00349`.

[34] E. Nasr-Esfahani, N. Karimi, S. M. R. Soroushmehr, M. H. Jafari, M. A. Khorsandi, S. Samavi, and K. Najarian, “Hand gesture recognition for contactless device control in operating rooms,” *CoRR*, vol. abs/1611.04138, 2016. `https://doi.org/10.48550/arXiv.1611.04138`.

[35] A.-r. Lee, Y. Cho, S. Jin, and N. Kim, “Enhancement of surgical hand gesture recognition using a capsule network for a contactless interface in the operating room,” *Computer Methods and Programs in Biomedicine*, vol. 190, p. 105385, 2020. `https://doi.org/10.1016/j.cmpb.2020.105385`.

[36] Z. Lu, S. Qin, X. Li, L. Li, and D. Zhang, “One-shot learning hand gesture recognition based on modified 3d convolutional neural networks,” *Machine Vision and Applications*, vol. 30, 08 2019. `https://doi.org/10.1007/s00138-019-01043-7`.