Jigsaws Soup: Improved Laparoscopic Surgical Gesture Classification, and Skill Level Evaluation Using Model Souping

Kade MacWilliams[a], Ahmed Nasr[b], Georges Azzie[c], James R. Green[a], Carlos Rossa[a]

[a]*Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada*
[b]*Division of Pediatric Surgery, Children's Hospital of Eastern Ontario, Ottawa, ON, Canada*
[c]*Department of Surgery, Hospital for Sick Children, Toronto, ON, Canada*

Abstract

Laparoscopy has revolutionised surgery, with faster recovery and less trauma for patients. However, extensive training is needed to gain the required skills, with said training ideally involving simulation-based strategies. A shortage of human expert coaches results in a lack of training opportunities in many jurisdictions. An Artificial Intelligence (AI) coach that can replicate the feedback from a human expert can be utilised to make quality surgical training more accessible. To be effective, this AI coach must classify complex movements into meaningful surgical gestures and then assess the trainee's performance on several parameters. We apply model soup, a recently developed machine learning approach, to improve gesture recognition and surgical score regression

[1]Email: kademacwilliams@cmail.carleton.ca (K. MacWilliams, Corresponding Author); jrgreen@sce.carleton.ca (J.R. Green); anasr@cheo.on.ca (A. Nasr); georges.azzie@sickkids.ca (G. Azzie); carlosrossa@cunet.carleton.ca (C. Rossa).

[2]This research is supported by the Queen Elizabeth II Scholarship in Science and Technology, the Canadian Graduate Scholarship from the Natural Sciences and Engineering Research Council of Canada, and the Children's Hospital of Academic Medical Organization (CHAMO) innovation fund.

[3]Cette recherche est appuyée par la bourse d'études supérieures en sciences et technologie de la reine Élizabeth II, par la bourse d'études supérieures Canadienne du Conseil de recherches en sciences naturelles et en génie du Canada, et par les fonds d'innovation CHAMO.

from kinematic simulator data. We also apply model soup to improve surgical gesture recognition from video data. By mixing several different 'ingredient' models into a final 'soup' model, performance is increased by 2.84-7.25 percentage points across three surgical tasks over state-of-the-art methods. These improved accuracies bring us closer to a fully autonomous laparoscopic surgery coach, with the potential to dramatically increase the availability of quality training, especially in environments where the availability of teachers and coaches is insufficient.

Keywords: Surgical training, Gesture recognition, Skill assessment, Model soup, Deep learning

---

1. Introduction

Laparoscopy is a form of Minimally Invasive Surgery (MIS) in which thin instruments and a camera are passed through small abdominal incisions, resulting in less trauma, faster recovery, and shorter hospital stays compared to open surgery [1, 2]. To complete a laparoscopic procedure, a surgeon must rely on a 2D video feed of a 3D operation, work with a limited field of view, lose haptic feedback, and sacrifice two degrees of freedom. Consequently, laparoscopy has a steep learning curve to develop the necessary cognitive and technical skills [3, 4]. Training methods range from intra-operative observation and cadaveric models to Virtual Reality (VR) simulators and box trainers [5, 6].

Along with traditional laparoscopy, Robotic-Assisted MIS (RAMIS) systems are commonly used in operating rooms. RAMIS involves utilising robotic arms and a control console to manipulate laparoscopic tools, rather than doing so by hand. RAMIS, while much more expensive than traditional laparoscopic surgery has several advantages, including motion-scaling, motion and tremor filtering, an additional degree of freedom at the tool tip, and a 3D endoscopic view [7]. However, similar to traditional laparoscopy, RAMIS requires extensive training to master the fine motor skills required to perform a MIS through a robotic control console. Systems such as the daVinci Skills Simulator (DVSS) allow surgeons to practice complex RAMIS procedures prior to entering the operating room. Work on the daVinci Research Kit (DVRK), which repurposes decommissioned daVinci Surgical Systems for research, allows for the collection of large amounts of high-quality kinematic and video data that can be used to create accurate models to assist in the

training and overall performance of surgeons on platforms such as the DVSS [8, 9].

Box trainers consist of a box with an internal camera and ports for inserting instruments, along with internal modules for training tasks. Drills performed in such simulators, such as bean drop and object manipulation, may improve performance in the operating room after 30-35 repetitions and over six months of practice [10, 11]. These simulators are the most widely used tools for laparoscopic training since they are affordable, offer tactile feedback, and are easy to deploy. However, they do not simulate variable workspace sizes, a notable limitation given that the abdominal cavity in paediatric patients can be 20 times smaller than that of adults [6]. While VR systems offer better visual realism and some are able to provide instant and automated performance feedback [12, 13], they are expensive and lack haptic feedback. Combining automated assessment with low-cost box trainers could bridge this educational gap and enhance training accessibility and consistency.

Performance evaluation during surgical training is usually a two-step process. First, the simulator determines which surgical gesture the trainee is attempting to perform. A surgical gesture is a specific movement within a segment of a surgical task, for example, reaching for a needle with the instrument. A surgical task refers to a specific sequence of such gestures, combined to achieve a larger surgical goal. Next, the simulator evaluates how well the trainee performed the gesture in relation to a reference standard. Usually, this involves identifying both the overall task and the surgeon's current position within it. Then the system can provide performance feedback on the individual gesture or the entire task.

Training devices often collect large amounts of kinematic and/or video data. Using such data, many algorithms have been developed to identify surgical gestures and tasks, including dynamic time warping, Markov models, Recurrent Neural Networks (RNNs), and Convolutional Neural Networks (CNNs) [14, 15, 16, 17]. Our previous work used a 1-D CNN to classify kinematic data sequences into specific surgical gestures [18]. Researchers have also fused predictions from a Multi-Modal Relational Graph Neural Network (MRG-Net) with a Residual Network (ResNet) CNN model for video data and a Long-Short Term Memory (LSTM) model for kinematic data [19]. Others have opted for video-based transformer models that use tool-tip kinematics to classify surgical gestures [20]. Vision transformers and supervised contrastive learning have been developed to identify and classify surgical gestures [21]. To identify surgical tasks from kinematic data,

researchers have used autoencoders with LSTMs, and [22] Hidden Markov Models (HMMs) [23].

While identifying surgical gestures and tasks is critical for assessing surgical performance, it offers no feedback to a trainee. A second set of algorithms is required to provide trainees with usable, actionable, and automated feedback based on the collected kinematic or video data. Such feedback algorithms can be grouped into two main categories: quantitative feedback (machine-centric) and qualitative feedback (human-centric).

Quantitative feedback in surgical training is usually given in the form of tool-tip path/trajectory, applied force, and deviation metrics compared to those of an expert surgeon. For example, previous algorithms have compared the Euclidean distance between the trainee and the expert surgeon's tool-tip position across a surgical task to classify the trainee's skill level [24]. However, converting complex kinematic data or post-processed video information from machine data into clear, actionable feedback that helps trainees improve surgical skills is challenging.

Qualitative feedback often grades a trainee's surgical performance by skill level (expert, intermediate, novice), Likert-scale scoring such as the Global Rating Scale (GRS) [25], or surgical safety scores such as the Surgical Apgar Score (SAS). Table 1 shows the GRS scoring system, where scores are attributed by an observing expert surgeon. While these metrics can provide more intuitive feedback, they rely on one-to-one supervision. An algorithm that accurately predicts a GRS score would allow for more autonomous training without the need for an observing expert surgeon. Rather than attempting to coarsely classify a trainee's performance (novice, intermediate, expert), predicting the GRS score can be formulated as a regression problem. For example, both CNN+LSTM models [26] and regression transformers [27] have been used to extract temporal sequences from the video and kinematic data to predict a GRS score. Several methods have also been developed to regress GRS directly from video data [28, 29].

Accurate classification and performance evaluation are crucial for effective surgical skill training, assessment, and creating automated and actionable feedback. Currently, the best performing models range from 75.68% to 92.56% in gesture classification, and a Spearman Correlation Coefficient (SCC) of 0.61 for GRS score regression [30]. The traditional machine learning approach is to select a model architecture, train individual models with varied hyperparameters, and then select the single model with the best performance, discarding all others. However, single models can perform sub-optimally on

Table 1: Elements of Modified Global Rating Score [25].

Element	**Rating Scale**
Respect for tissue	1 – Frequently used unnecessary force on tissue 3 – Careful handling but with occasional tissue damage 5 – Consistent appropriate tissue handling
Suture/needle handling	1 – Awkward, repeated entanglement and poor tying 3 – Most knots placed correctly with tension 5 – Excellent suture control
Time and motion	1 – Made unnecessary moves 3 – Efficient time/motion, some unnecessary moves 5 – Economy of movement and maximum efficiency
Flow of operation	1 – Often interrupted flow to discuss the next move 3 – Demonstrated planning and procedure progression 5 – Obviously planned with efficient move transitions
Operation flow/planning	1 – Frequently stops and hesitations 3 – Forward planning with steady progression 5 – Planned course of operation with effortless flow
Knowledge of procedure	1 – Deficient knowledge 3 – Familiarity with important aspects of the procedure 5 – Familiarity with all aspects of the procedure

Out-Of-Distribution data (OOD) and result in poor performance on test data. A common approach to improve the performance of single-model architectures is to use ensemble models. Ensemble models combine the outputs of multiple models to improve performance. However, ensemble models are computationally expensive with a high inference time, since the entire collection of trained models must be run at inference time.

A potential solution to efficiently merge the outcomes of different models is "model souping" [31]. Model soup involves individually training separate models of the same architecture and evaluating their performance. All models (soup ingredients) in the soup are fine-tuned versions of the same original model (stock/pre-trained), each fine-tuned with slightly different hyperparameters. This permits exploration of the solution space surrounding the original stock model. The final model represents the average of the top-performing ingredient models. [31]. The souped model is then used without further tuning or training. Model soup was recently proposed in [31] with improved results on ImageNet classification over single and ensemble models.

We hypothesize that model soup will allow for increases in performance in gesture recognition and expertise regression, as it has been shown to perform well with both In-Distribution (ID) and OOD data [31]. It offers ensemble-like

performance without increasing the inference time. Recently, model soup has been used in various biomedical applications, such as Electroencephalogram (EEG) analysis and interpretation, skin cancer image classification, and vertigo detection from images of a user's eyes [32, 33, 34]. Model soup has also been shown to work well in zero-shot learning applications, where the model is evaluated on classes that it has not been explicitly trained to recognise [31]. We expect that this ability can be advantageous in surgical applications; for example, model soup could be used to learn a model from RAMIS while being deployed on traditional laparoscopic training devices, with limited fine-tuning of the model. This may also result in models that have strong adaptation abilities to very specific tasks, such as paediatric laparoscopic surgical classification and evaluation.

In this paper, we demonstrate for the first time that model soup improves classification and assessment accuracy of surgical gestures, tasks, and GRS-based skill level prediction. Validated using the Johns Hopkins University and Intuitive Surgical Inc Gesture and Skill Assessment Working Set (JIGSAWS), the model shows an improvement of 7.25 percentage points in surgical gesture classification using only kinematic data, and a significant increase of 0.09 points (0.61 to 0.70) on the SCC for GRS score prediction over the state-of-the-art. To the best of our knowledge, this is the first application of model soup to surgical gesture classification and GRS score prediction. A description of the soup algorithm is presented in Section II, followed by the architectures used for both classification and performance evaluation in Section III. The JIGSAWS dataset structure and testing protocol for the algorithms are presented in Section IV. Section V presents the results of each model in the context of several other state-of-the-art algorithms. Finally, Section VI will serve as the conclusion of this work and present thoughts on future development in this field.

2. Model Soup

Model soup is a method for combining the learned and optimised weights of multiple models. It creates a new single (souped) model that often generalises and performs better than any individual model from which it was derived [31], while maintaining the inference time of a single model. A model soup is made by first selecting a (stock) model architecture and pretraining it with fixed hyperparameters. The learned weights from this pretrained stock model serve as the starting weights for fine-tuning multiple (ingredient) models with

different hyperparameters. The fine-tuned weights of these ingredient models are then combined using a "soup recipe" to produce a new single (soup) model. The overall workflow of this process is illustrated in Fig. 1.

The general recipe for making a model soup is as follows:

1. Select a model architecture;
2. Pretrain the model with fixed hyperparameters;
3. Use the obtained weights as the starting weights of k new models (soup "ingredients") of the same architecture;
4. Fine-tune each of the ingredient models with slightly different hyperparameters;
5. Combine soup ingredients following one of the soup recipes.

2.1. Pretrained Model and Soup Ingredients

After pretraining a model with fixed hyperparameters, the learned model weights, w_{pt}, will serve as the starting point for k new models. These models (hereafter referred to as a "soup model") will have the same architecture as the pretrained model, but each of them has different hyperparameters. Some of the hyperparameters that can be varied between soup models include learning rate, dropout rate, number of epochs, etc.

Fine-tuned models that start from the same pretrained weights typically lie within the same loss basin, which allows the problem of combining the weights to be treated as convex [35]. Model soup aims to find the flat centre of this loss basin since weights in this region generally result in better model generalisation [31, 35]. After each ingredient model, i, is fine-tuned, its learned model weights are w_i, resulting in a set of weights $W = \{w_1, w_2, \ldots, w_k\}$. The validation accuracy, A_i, of each fine-tuned ingredient model is also recorded.

2.2. Soup Recipe

To create a model soup, several sets of input data are required. A pretraining dataset is required to train the pretrained/fixed model to find starting weights for our ingredient models. A fine-tuning dataset is also required to tune the individual ingredient models. A subsection of this data must be held out for validation, while another subsection of the data must be held out for the final evaluation (testing) of the soup model.

There are two main varieties (recipes) of model soups: greedy soups and uniform soups.

2.2.1. Uniform Soup

A uniform soup is created by taking all k soup models and uniformly averaging their learned weights, w_i, resulting in final model weights w_m. Once the average is calculated, a final model is made using the averaged weights without any further training or changes required.

$$w_m = \frac{1}{k} \sum_{i=1}^{i=k} w_i \tag{1}$$

2.2.2. Greedy Soup

A greedy soup ensures that the final model performs as well as the best-performing soup model on the validation set. Unlike a uniform soup, a greedy soup only selects ingredients that improve the model's accuracy on the held-out validation set.

Considering the accuracy again, A_i, of soup model i, on the held-out validation set, the greedy soup is made as follows:

- Sort ingredient models in descending order according to their validation accuracies, A_i. Let w_m be the weights of the model with the highest validation accuracy;
- Set the final model weights, w_g, to the weight of the highest performing soup model, that is $w_g = w_m$;
- Select the next best remaining model, and average its weights with z (where z is the number of models' weights included in $w_m + 1$), best performing weights, w_m, such that

  $$w_{a_i} = \frac{1}{z}(w_m + w_i)$$

  and calculate the resulting validation accuracy A_{a_i} using w_{a_i} in the final model;
- If $A_{a_i} > A_m$, set $w_m = w_{a_i}$
- Otherwise, remove soup model weight w_i from the pool of remaining ingredient models and repeat the process until no soup model weights w_i remain untested.

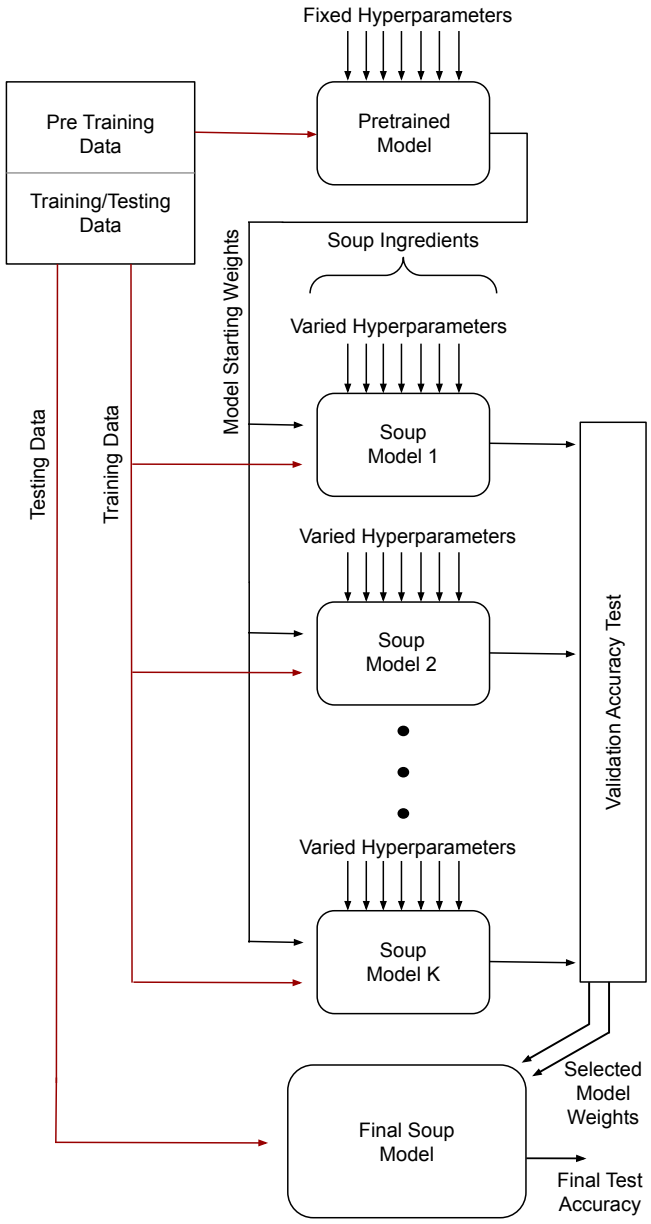


Figure 1: A general overview of the model soup algorithm used in this paper. The pretrained model weights are used as starting weights for all soup models. The soup ingredient models' learned weights are then combined following the greedy recipe. The final soup model takes the recipe weights as the learned weights.

This recipe guarantees that the final weights w_{a_i} yield a higher validation accuracy than any single soup model weight. Once the process is completed, set the final model weights $w_g = w_m$. The greedy soup recipe is also summarised in pseudo-code below.

Algorithm 1 Greedy Soup Recipe [31]

Input: Potential Soup Ingredient Model weights $w_1, \ldots, w_k$ sorted in decreasing order of A_i.
Ingredients $\leftarrow \{\}$
for $i = 1$ **to** k **do**
  if $A(\text{average}(\text{Ingredients} \cup \{w_i\})) \geq A(\text{average}(\text{Ingredients}))$
    then Ingredients $\leftarrow$ Ingredients $\cup \{w_i\}$
return average(Ingredients)

Both greedy and uniform recipes have memory and compute costs equal to those of a single model at inference time. This is a major benefit over ensemble methods, which have memory and compute costs that scale proportionally to the number of component models in the ensemble. Additionally, soup models still benefit from stronger generalisation, similar to ensemble methods, due to the interpolation of learned weights in parameter space.

3. Soup Models for Gesture, Task and Proficiency Classification

The goal of this work is to investigate whether model soup benefits gesture classification and GRS score regression using kinematic data. We use a 1D-CNN as a common model backbone for both gesture recognition and GRS score prediction, with the loss functions and classification/regression heads differing. In both cases, the following CNN model hyperparameters are varied during fine-tuning of ingredient models: 1) Number of epochs, 2) Learning rate, 3) Batch size, 4) Dropout, and 5) Weight decay.

Hyperparameter configurations were randomly selected within a bounded range for each parameter.

For GRS score regression, the hyperparameters were varied as follows: learning rates were between $1e^{-4}$ to $5e^{-5}$, the batch size was either 1 or 2, the number of epochs varied between 15 to 30, the weight decay varied between 0 to $5e^{-5}$, and the dropout rate varied between 0 to 0.5.

For kinematic gesture recognition, the hyperparameters were varied as follows: learning rates were between $1e^{-4}$ to $5e^{-5}$, the batch sizes were either

8 or 16, the number of epochs varied between 10 to 30, the weight decay varied between 0 to $5e^{-5}$, and the dropout rate varied between 0 to 0.5.

3.1. Gesture Classification and GRS Score Prediction Models

The model architecture used for gesture classification with the kinematic data is a 1-D CNN. The greedy soup recipe described in Section II is used to create the final soup model from 10 ingredient soup models.

The 1-D CNN accepts samples in the form of discrete temporal sequences. Each input sequence is zero-padded such that all samples have a length of ℓ elements. Each sample has c dimensions corresponding to different kinematic measurements, such as the position or speed of the surgical tool in one of three dimensions. Each kinematic variable is standardised during pre-processing across all training sequences. This manipulation prevents one feature with larger numerical values from dominating the model training.

Let $\mathbf{X}^s \in \mathbb{R}^{c \times \ell}$ be a matrix containing the kinematic data of a sample s, with $1 \leq s \leq N$, where N is the number of samples in the database. Also, let X^s_{pq} be the element in the p^{th} column and q^{th} row of $\mathbf{X}^s$. Each column $1 \leq p \leq c$ of $\mathbf{X}_{pq}$ contains a dimension (feature) of the kinematic data, and each row $1 \leq q \leq \ell$ is a time step. The mean μ_p of a given dimension p across all samples can be computed as:

$$\mu_p = \frac{1}{N\,\ell} \sum_{s=1}^{N} \sum_{q=1}^{\ell} X^s_{pq} \tag{2}$$

and the standard deviation σ_p of the the same dimension p is

$$\sigma_p = \sqrt{\frac{1}{N\,\ell} \sum_{s=1}^{N} \sum_{q=1}^{\ell} \left(X^s_{pq} - \mu_p\right)^2 \; + \; \epsilon}, \tag{3}$$

where $\epsilon = 10^{-6}$ is an offset that ensures $\sigma_p \neq 0$. To enforce that all samples $\mathbf{X}^s$ have a mean of $\mu_p = 0$ and a standard deviation of $\sigma_p = 1$, we standardize each sample such that

$$X^{s'}_{pq} = \frac{X^s_{pq} - \mu_p}{\sigma_p}. \tag{4}$$

The standardised kinematic data is then input into the 1-D CNN model. The backbone of the 1-D CNN is structured as follows:

1. Convolution layer:
   Filters: 32, Kernel Size: 5, and Padding: 2
2. ReLU activation layer
3. Max pooling layer: 2 Kernels
4. Convolution layer:
   Filter: 64, Kernels: 5, and Padding: 2
5. ReLu Activation Layer
6. Max pooling layer: 2 Kernels
7. Flatten layer producing a vector of length 64×(Length/4)
8. Fully connected dense layer with 128 units
9. ReLu activation layer

   GRS Regression Head:
10. Dropout layer with probability b
11. Dense layer with 1 unit for GRS regression

    Surgical Gesture Classification Head:
12. Dropout layer with probability h
13. Fully connected layer mapping 128 to n gesture classes

This backbone is the same for both gesture recognition and GRS regression. Only the output layer and loss function change between a multi-class classification head and a regression head.

Surgical Gesture Classification: Surgical gesture classification assigns each trial, $\mathbf{X}_s$, one of n possible gesture labels (classes) L. A 1-D CNN with a classification head produces n raw output values (logits), each $z_i \in \mathbb{R}$. These logits are converted to probabilities, and the class with the highest probability is selected as the predicted surgical gesture. Training uses the multiclass cross-entropy loss.

Global Rating Scale Regression: To assign a GRS score to each sample $\mathbf{X}^s$ a regression model is used. Model evaluation measures both the Mean Absolute Error (MAE) and the Mean Squared Error (MSE) between a predicted GRS score $\hat{L}^s$ and the true GRS score L^s of $\mathbf{X}^s$ of m samples:

$$MAE = \frac{1}{m}\sum_{s=1}^{m} \hat{L}^s - L^s, \qquad MSE = \frac{1}{m}\sum_{s=1}^{m} \left(\hat{L}^s - L^s\right)^2 \tag{5}$$

An MSE loss function (5) is used during 1-D CNN training, which compares the output logit to the true GRS score for a given sample, $\mathbf{X}^s$. Using an MSE loss function provided more accurate GRS score predictions than using an MAE loss function. MAE given in (5) is used for ingredient model ranking during assembly of the final soup, using the greedy soup recipe. MAE is used to compare our greedy soup method to other single model and uniform soup model methods, as this is the best metric to summarise the error between $\hat{L}^s$ and L^s.

4. Dataset and Data Preprocessing

To implement the proposed algorithms, we use the publicly available Johns Hopkins University and Intuitive Surgical Inc. Gesture and Skill Assessment Working Set (JIGSAWS) [14]. This dataset includes both video recordings and 76 kinematic signals from eight surgeons performing three surgical tasks (suturing, needle passing, and knot tying) on the daVinci laparoscopic Surgical System. Of the eight surgeons who participated in the study, two self-identified as experts (<100 hours experience), two as intermediates (10-100 hours), and four as novices (<10 hours).

Each surgical task is broken down into a sequential combination of 15 possible annotated surgical gestures, as listed in Table 2. In knot tying, the surgeon picks up one end of a suture, while the other end is tied to a flexible tube. The surgeon ties a single loop knot. During suturing, the surgeon picks up a needle and passes it through the tissue three times. Finally, during needle passing, the surgeon picks up a needle and passes it through small metal hoops from right to left [14]. Each surgeon performed each of these three tasks five times, although the data from some repetitions were found to be corrupted and could not be used in our study.

In total, 39 trials of suturing, 36 trials of knot tying, and 28 trials of needle passing are used, with the remaining trials being discarded due to data corruption. In total, the dataset includes 1703 total gestures, as seen in Tab. 2. In addition to self-reporting of expertise, the JIGSAWS dataset also assigns a GRS score to each task trial. This score was given by a gynecologic surgeon with extensive robotic and laparoscopic surgical experience [14]. These GRS scores are used to develop and evaluate our kinematic-based GRS regression models.

The kinematic data contains 76 total dimensions; however, only 38 dimensions (master side of the daVinci) of the data are used to train and test

Table 2: List of gestures and tasks used in the evaluation, including what gestures compose a task, and the number of samples for each in the database [14].

Gesture	**Description**	Samples
G1	Reaching for needle with right hand	78
G2	Positioning needle	283
G3	Pushing needle through tissue	275
G4	Transferring needle from left to right	202
G5	Moving to centre with needle in grip	68
G6	Pulling suture with left hand	275
G7	Pulling suture with right hand	0
G8	Orienting needle	76
G9	Using right hand to help tighten suture	25
G10	Loosening more suture	5
G11	Dropping suture and moving to end point	100
G12	Reaching for needle with left hand	70
G13	Making C loop around right hand	75
G14	Reaching for suture with right hand	98
G15	Pulling suture with both hands	73
Task	**Gestures in Task**	
Suturing	G1-6, G8-11	39
Knot tying	G1, G11-15	36
Needle passing	G1-6, G8, G11	28

our models. This decision to use only the master side data was made to help reduce the dimensionality of input data, especially considering that all kinematic data in the discarded data are just some scaled version of the master side kinematic data.

Leave-One-User-Out (LOUO) and Leave-One-Supertrial-Out (LOSO) cross-validation were used, with the average accuracy or mean average error reported, depending on the classification activity. LOUO cross-validation involves leaving out all of the data from one of the eight surgeons during model training and validation, while reserving their data exclusively for testing the final model. LOSO cross-validation is less conservative than LOUO, where only a single attempt of a surgical task from each surgeon is withheld for testing, such that a model may be trained and tested on data from the same

surgeon, but not the same surgical task attempt.

Under both LOUO and LOSO cross-validation, there will be three sets of partitioned data: a training set, a validation set, and a test set. The test set comes from the held-out data under the LOUO or LOSO test scheme. For gesture recognition models, an 80-20 training/validation split was used. For GRS prediction models, a 70-30 training/validation split was used. In both cases, the split was done simply across all available samples, not by surgeon or any other method. The training set is used to fine-tune each of the potential soup ingredients, and the validation set is used to test/select ingredients for the final soup, following the greedy soup recipe.

The three surgical tasks each formed a dataset, consisting of all valid trials of that task. The tasks were divided into known gestures using the provided gesture labels. The soup model requires pretraining to establish starting model weights, as explained in Section II. The starting pretrained model is trained on two of the JIGSAWS tasks (supervised learning, using the gesture or GRS label data), with the third task being reserved for fine-tuning each of the potential soup ingredients and testing. The testing scheme outlined above (LOSO or LOUO) is strictly enforced on this third task, with the testing data being completely removed from the training set to ensure no data leakage.

To evaluate our models, following the JIGSAWS benchmark paper [14], several statistical summary statistics are reported for both gesture classification accuracy and GRS score regression. These statistics include macro and micro averages, macro standard deviation, average precision, precision standard deviation, β distributions ($\mu \pm \sigma$), and β 95% confidence interval for gesture recognition. For GRS score prediction, we report the MAE, as well as the SCC.

4.1. Gesture Recognition Evaluation Metrics

Given a confusion matrix $C(i, j)$ with $1 \leq i, j \leq n$, where n is the number of classes, the micro-average $\bar{x}_{mi}$ is computed as the total number of correct predictions over the total number of predictions:

$$\bar{x}_{mi} = \frac{\sum_{i=1}^{n} C[i, i]}{\sum_{i,j=1}^{n} C[i, j]}. \tag{6}$$

The macro-average $\bar{x}_{ma}$ is computed as the mean of the true positive rate for each class:

$$\bar{x}_{ma} = \frac{1}{n} \sum_{i=1}^{n} \frac{C[i, i]}{\sum_{i=1}^{n} C[i, j]} \tag{7}$$

and the macro average's standard deviation is:

$$\sigma_{ma} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{C[i,i]}{\sum_{j=1}^{n} C[i,j]} - \bar{x}_{ma}\right)^2}. \tag{8}$$

Precision P_r and precision standard deviation σ_P represent the proportion of true positives to predicted positives and are calculated as:

$$P_r = \frac{1}{n}\sum_{i=1}^{n}\frac{C[i,i]}{\sum_{k=1}^{n} C[k,i]} \tag{9}$$

$$\sigma_P = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{C[i,i]}{\sum_{k=1}^{n} C[k,i]} - P_r\right)^2}. \tag{10}$$

4.2. GRS Score Prediction Evaluation Metrics

The MAE is used to evaluate the difference between a predicted GRS score $\hat{L}^s$ and the true GRS score L^s of $\mathbf{X}^s$ of m samples

$$MAE = \frac{1}{m}\sum_{s=1}^{m}\hat{L}^s - L^s, \tag{11}$$

To confirm that our predicted scores correlate with actual scores, we use the Spearman Correlation Coefficient (SCC).

$$\rho = 1 - \frac{6\sum_{s=1}^{m} L^s - \hat{L}^s}{m(m^2-1)} \tag{12}$$

The SCC compares all ground truth GRS scores, L^s, to our predicted GRS scores, $\hat{L}^s$.

5. Results and Discussion

We compare the performance of greedy and uniform soups against the best single ingredient and several SOTA baseline methods, for both gesture classification and GRS prediction. In both cases, kinematic data are used as input to all models.

To ensure robustness of our method, we also compute various statistics collected from 100 random seed trial runs, including: the expected micro-average, $\bar{x}_{mi}$, expected precision, P_r, and the 95% confidence interval for each of these statistics.

Our GRS regression model was compared with several state-of-the-art methods, as outlined in Table 3. Zia and Essa have used the Discrete Cosine Transform (DCT)-Discrete Fourier Transform (DFT)-Approximate Entropy (ApEn) to predict a GRS score [30]. These three features are extracted from the kinematic data, and the dimensionality of the data is reduced using Principal Component Analysis (PCA) before GRS score prediction. The SCC is reported for each predicted GRS score.

The Recursive Cross-Attention Network for Pseudo-Label Generation (ReCAP) method has been proposed in [27], and uses segments of kinematic data to predict intermediate scores, which are then averaged at the end of the trial to produce an overall score. The results in this paper show an MAE for suturing, knot tying, and needle passing of 2.89, 2.04, and 3.12 using LOSO cross-validation, respectively. However, our method cannot be compared to this work, as it is stated in [27] that the model hyperparameters were tuned to the test fold. The ReCAP method uses a Recurrent Neural Network (RNN) that fuses the previous hidden state of the recurrent network with the current segment-level kinematic signal. Self-attention is applied to each prior to fusion, which is then followed by cross-attention. Each hidden state is then passed to six classification heads, which predict a score. The overall score is the average result from all segment-level predictions.

For GRS regression with the greedy soup 1D-CNN model, we obtained an MAE of 2.98 and an SCC of 0.70 on LOSO cross-validation methods, respectively. These results, summarised in Tables 3 and 4, show that the proposed greedy model soup algorithm consistently outperforms both the single and uniform soup models in GRS score regression when using kinematic data. The best single model used in the soup had an average MAE of 3.33 and SCC of 0.61, which lowered the MAE to 2.98 and raised the SCC to 0.70 after souping with other models. As can be seen in Table 4, our greedy model soup also achieves an SCC 0.09, which is higher than the best method listed in [30].

The average SCC values, for the three tasks, for LOSO and LOUO are, respectively, 0.7 and 0.2. The SCC measures how well the ranking of predicted GRS scores matches the ranking of true GRS scores for all samples in a test set. A single LOUO cross-validation fold contains all the data from a single

Table 3: Comparison of Mean Average Error (MAE) of GRS regression for all tasks using kinematic data (lower is better).

Data	Method	Suturing	Needle passing	Knot tying	Avg.
	LOSO				
Kinematic	Uniform Soup-CNN (this paper)	3.70	4.03	2.71	3.48
Kinematic	Single CNN (this paper)	3.52	3.64	2.84	3.33
Kinematic	Soup-CNN (this paper)	**3.34**	**3.27**	**2.34**	**2.98**
	LOUO				
Kinematic	Uniform Soup-CNN (this paper)	5.52	4.32	3.34	4.39
Kinematic	Single CNN (this paper)	4.90	4.08	3.54	4.17
Kinematic	Greedy Soup-CNN (this paper)	**4.38**	**3.63**	**2.71**	**3.57**

surgeon, and as a result, the true GRS scores have a lower range of possible GRS scores within each test set. The model must also regress a GRS score without being allowed to see any data from that surgeon, unlike under LOSO cross-validation. With a narrower range, small prediction errors in GRS scores have a large impact on the ranking, lowering the SCC. A LOSO cross-validation includes a trial from each surgeon, with different skill levels, allowing the model to see the remaining trials from that surgeon. As a result, the true GRS scores have a larger range. Small errors in predicting GRS scores have a limited impact on the SCC. However, when the predicted LOUO GRS scores are evaluated using SCC across an entire trial, rather than as the average of single folds, the SCC ranges from 0.42 to 0.61 across the three tasks. This verifies that the discrepancy comes from the more narrow range of possible GRS scores in a single fold of LOUO cross-validation, and the fact that the model has not seen any data from the tested surgeon, unlike under LOSO cross-validation.

The second half of the table shows the results of a single model and the model soup with LOUO validation. LOUO is generally considered a more conservative evaluation method than LOSO as it excludes all data from one surgeon during model training and tests, while LOSO only excludes one sample or trial per surgeon. However, we found no other papers in the literature that reported GRS score regression accuracy with LOUO cross-validation and kinematic data only. The soup model had an average LOUO MAE of 3.57, compared to a strong-performing single-ingredient model, which had an average MAE of 4.17. This shows that soup improves the MAE over its constituent ingredients.

Table 4: Comparison of Spearman Correlation Coefficient (SCC) of GRS regression for all tasks using kinematic data. (Closer to a value of 1.0 is better)

Data	Method	Suturing	Needle passing	Knot tying	Avg.
	LOSO				
Kinematic	DCT-DFT-ApEn [30]	**0.75**	0.46	0.63	0.61
Kinematic	Single-CNN (this paper)	0.62	0.49	0.74	0.61
Kinematic	Uniform Soup-CNN (this paper)	0.63	0.56	0.72	0.64
Kinematic	Soup-CNN (this paper)	0.70	**0.63**	**0.78**	**0.70**
	LOUO				
Kinematic	Single CNN (this paper)	**0.22**	0.03	0.019	0.09
Kinematic	Uniform Soup-CNN (this paper)	0.18	0.09	0.08	0.12
Kinematic	Greedy Soup-CNN (this paper)	0.12	**0.37**	**0.22**	**0.23**

We compared our gesture recognition results to several benchmark methods in [36]. The Linear Dynamical System (LDS) models each kinematic frame as the output of an LDS with Gaussian noise as an input [36]. For every pre-segmented sequence, an LDS is fitted using a PCA-based identification method, producing a set of system parameters for each segment. A separate LDS is fitted to each manually segmented gesture sequence, so that every training example is represented by its own dynamical model. Pairwise distances between LDS models are then computed using several metrics, including Martin, Frobenius, Binet–Cauchy, and alignment distances. These distances are used to train a classifier, typically k-Nearest Neighbours (k-NN) or an RBF-kernel Support Vector Machine (SVM), to assign gesture labels to kinematic data. The Gaussian Mixture Model (GMM) - Hidden Markov Model (HMM) implemented in [36] modelled each gesture as an elementary HMM where each state corresponds to one GMM. The elementary HMMs for different gestures are then concatenated to form a composite HMM that models the entire trial.

The results for gesture classification using kinematic data are summarised in Tables 5, 6. Model soup improves the performance of gesture classification under both LOUO and LOSO compared to the benchmark paper [36]. Improvement over the benchmark methods is seen in every category of kinematic surgical gesture recognition. Under the LOSO test scheme, the greedy soup-CNN achieved an average micro accuracy of 88.84%, compared to 86% micro accuracy from the best benchmark method. The greedy soup model also outperformed a single-ingredient model by an average of 4.32 percentage points and the uniform soup model by an average of 9.33 percentage points,

Table 5: Comparison of gesture recognition methods from kinematic data using LOUO for gestures within suturing, needle passing, and knot tying. LDS is a Linear Dynamical System, and GMM-HMM is a Gaussian Mixture Model-Hidden Markov Model.

Method (Data type)	**Metric**	**Suturing**	**Needle passing**	**Knot tying**
		LOUO		
LDS [36]	Micro	73.64	47.96	71.42
	Macro±std	51.75±32.91	32.59±29.74	63.99±24.51
	Precision±std	53.39±32.01	32.01±27.76	65.74±21.54
	β ($\mu \pm \sigma$)	73.80±0.300	45.68±1.450	71.44±0.15
	β 95%CI	62.31–83.84	22.92–69.44	63.31–78.93
GMM-HMM [36]	Micro	80.83	66.22	78.44
	Macro±std	65.03±33.07	62.70±16.38	72.68±21.31
	Precision±std	71.98±33.05	66.55±25.45	83.27±13.04
	β ($\mu \pm \sigma$)	81.31±01.32	69.19±2.907	78.16±0.508
	β 95%CI	54.16–97.47	31.31–95.59	61.47–91.07
Single-CNN This paper	Micro	81.08	74.76	77.53
	Macro±std	62.67±35.15	65.63±21.76	76.14±9.24
	Precision±std	71.61±30.34	66.10±16.31	74.85±10.17
	β ($\mu \pm \sigma$)	81.98±10.48	74.76±7.87	77.53±8.77
	β 95%CI	57.38–97.06	57.93–88.42	58.23–92.04
Uniform Soup-CNN This paper	Micro	81.21	69.83	68.19
	Macro±std	61.31±36.36	60.00±28.40	72.26±21.51
	Precision±std	71.10±21.93	52.10±38.12	42.61±39.64
	β ($\mu \pm \sigma$)	82.05±10.14	70.83±11.87	68.44±11.37
	β 95%CI	58.35–96.82	45.22–90.71	44.34–88.11
Greedy Soup-CNN This paper	Micro	**85.00**	**78.40**	**83.83**
	Macro±std	65.60±36.87	71.51±19.08	82.76±7.67
	Precision±std	74.48±29.27	73.71±13.71	81.71±10.49
	β ($\mu \pm \sigma$)	85.47±7.74	79.67±9.65	83.47±8.98
	β 95%CI	67.35–96.88	57.82–94.75	62.53–96.70

under the LOUO test scheme.

Table 6: Comparison of gesture recognition methods from kinematic data using LOSO for gestures within suturing, needle passing, and knot tying. LDS is a Linear Dynamical System, and GMM-HMM is a Gaussian Mixture Model-Hidden Markov Model.

Method (Data type)	**Metric**	**Suturing**	**Needle passing**	**Knot tying**
		LOSO		
LDS [36]	Micro	84.61	59.77	81.67
	Macro±std	63.87±30.82	46.55±25.81	74.51±23.73
	Precision±std	73.30±28.41	52.91±17.31	76.07±18.72
	β ($\mu \pm \sigma$)	84.77±0.060	59.78±0.010	81.67±0.016
	β 95%CI	79.36–89.49	57.42–62.12	79.12–84.09
GMM-HMM [36]	Micro	92.56	75.68	89.76
	Macro±std	79.66±29.85	72.36±16.99	87.29±12.76
	Precision±std	81.20±30.42	73.60±20.43	91.52±7.410
	β ($\mu \pm \sigma$)	92.76±0.070	75.66±0.010	89.71±0.430
	β 95%CI	86.38–97.24	73.34–77.94	73.72–98.56
Single-CNN This paper	Micro	91.93	78.21	83.02
	Macro±std	78.87±29.83	68.59±21.05	79.30±14.00
	Precision±std	89.70±6.14	71.99±13.17	80.33±10.06
	β ($\mu \pm \sigma$)	92.47±6.03	78.14±14.33	83.16±13.98
	β 95%CI	77.05–99.53	48.07–99.43	63.81–92.47
Uniform Soup-CNN This paper	Micro	86.25	76.04	77.63
	Macro±std	65.56±39.01	47.64±44.69	78.32±6.76
	Precision±std	89.70±6.14	41.14±44.07	56.90±36.51
	β ($\mu \pm \sigma$)	92.47±6.03	79.70±17.33	77.76±16.66
	β 95%CI	77.05–99.53	36.73–99.56	37.82–98.90
Greedy Soup-CNN This paper	Micro	**93.06**	**83.42**	**90.03**
	Macro±std	79.17±30.39	77.24±18.01	90.43±4.17
	Precision±std	89.49±6.98	81.10±11.26	88.57±5.86
	β ($\mu \pm \sigma$)	93.60±6.07	83.20±16.62	90.00±15.21
	β 95%CI	86.8–97.9	86.83–94.71	80.31–86.56

5.1. Results from 100 Random Seed Trials

To test the robustness of our model, we conducted 100 random seed trials. Each random seed trial saw the entire creation/testing of a model soup. The random seeds influenced the following factors in the model simulation:

- Weight initialisation of the pretrained, stock CNN model
- Randomness in pretraining, including in data splits, and dropout masks
- Randomness in data shuffling performed by the data loader
- Randomness training and validation data split
- Introduces random variation into the gradient during training

The 100 random seeds were assigned from the corresponding trial number where, $Seed_i = Trial_i$.

For gesture recognition, the average micro-accuracy was reported, along with the 95% confidence interval, calculated following the beta distribution. The average precision across the 100 trials is also reported, with a 95% confidence interval calculated, also following the beta distribution. For GRS score prediction, the average MAE is computed across the 100 trials, with the 95% confidence interval calculated following a normal distribution. The average SCC is also calculated, with the 95% confidence interval calculated following a normal distribution. As can be seen in Tables 5 and 6, the reported values in the comparison portion of Section 5.1 fall well within the 95% confidence intervals. These results show that model soup increases the classification accuracy of surgical gestures and provides more accurate GRS scores for surgical performance evaluation.

Table 7: Expected Accuracy and Precision, along with 95% confidence interval (in brackets) from 100 random seed trial runs for suturing, needle passing, and knot tying gestures using LOUO and LOSO cross-validation.

Metric	**Suturing**	**Needle Passing**	**Knot Tying**
		LOUO	
Accuracy	0.8330 (0.7544–0.8991)	0.7589 (0.6709–0.8372)	0.8067 (0.7242–0.8776)
Precision	0.6954 (0.6021–0.7812)	0.7088 (0.6165–0.7932)	0.7546 (0.6661–0.8334)
		LOSO	
Accuracy	0.9181 (0.8573–0.9632)	0.8222 (0.7419–0.8903)	0.8788 (0.8084–0.9349)
Precision	0.8713 (0.8233–0.9455)	0.7922 (0.7079–0.8655)	0.8313 (0.7524–0.8977)

As can be seen in Tables 7 and 8, the reported values in the comparison portion of Section 5.1, fall within the 95% confidence intervals making the results reported not dependent on the initial seed value.

Table 8: Expected Mean Average Error (MAE) and Spearman Correlation Coefficient (SCC), along with 95% confidence Intervals (in brackets) from 100 random seed trial runs for suturing, needle passing, and knot tying gestures using LOUO and LOSO cross-validation.

Metric	Suturing	Needle Passing	Knot Tying
		LOUO	
MAE	5.32 (4.38–6.29)	4.03 (3.48–4.75)	3.10 (2.67–3.66)
SCC	0.15 (0.007–0.37)	0.16 (0.02–0.43)	0.08 (0.004–0.24)
		LOSO	
MAE	3.63 (3.22–4.06)	3.62 (3.07–4.32)	2.59 (2.27–2.95)
SCC	0.67 (0.49–0.68)	0.58 (0.42–0.69)	0.76 (0.70–0.82)

5.2. Number of Potential Ingredients in Greedy Soup

A sensitivity analysis was conducted to determine the effect of the number of potential soup ingredient models, k, on the model soup performance. It has been noted that model soups can begin to saturate and decrease in performance as an optimal number of models is exceeded [31]. The trials are shown in Table 9, differing only in the number of potential ingredients. As can be seen in Table 9, ten potential ingredient models yield the best performance in both gesture recognition and GRS score prediction. This finding motivates our use of 10 potential ingredients for the comparison portion.

Table 9: Effect of the number of potential ingredients, k, on a greedy soup for GRS score prediction (MAE) and gesture recognition (micro-accuracy) on the knot tying task using LOUO cross-validation. All trials are calculated with the same random seed.

k	MAE	Micro-Accuracy
2	2.94	81.40
5	2.84	78.98
10	**2.71**	**83.83**
15	2.95	80.05
20	2.94	79.25
50	3.35	79.51

These results show that a greedy model soup increases the classification accuracy of surgical gesture recognition and provides more accurate GRS scores for surgical performance evaluation. The performance exhibited by greedy soups exceeds not only uniform soup models and single models of the same architecture, but also current SOA methods from the literature.

6. Conclusion and Future Work

This paper explores model soup as a method to improve single model performance on a surgical gesture classification and GRS evaluation. The model soup algorithm offers the benefits seen in ensemble methods, such as increased performance on test data over a single model method, while retaining the cost of only a single model at inference time. These characteristics make model soup particularly relevant in surgical training algorithms, as the model may be able to generalise between surgeons of various skill levels, while remaining lightweight to be deployed on surgical training devices with limited computational resources.

Future investigation should examine the benefits of model soup using video and kinematic-video fusion data inputs. As model soup has been shown to boost the performance of image classification algorithms in [31] and video classification algorithms in [37], this machine learning methodology is a promising candidate for surgical gesture recognition and performance evaluation via video data. However, the findings of the present study are limited exclusively to kinematic data.

Model soup showed improvements in surgical gesture recognition and evaluation over benchmark methods. GRS score regression using a greedy soup approach saw a 14.4% improvement in MAE compared to either a single model or the uniform soup method, when using a LOUO testing scheme. Kinematic surgical gesture recognition also improved using the model soup strategy with an average increase of 7.25 percentage points in accuracy under the LOUO test scheme and an average increase of 2.84 percentage points under the LOSO test scheme, when compared to the best-performing benchmark methods.

Future validation is required to test the ability of the model to generalise to new robotic laparoscopic surgery datasets, where equipment and environmental factors may differ. Our use of LOUO testing does demonstrate that the model remains performant on previously unseen users. We are hopeful that future work will demonstrate successful generalisation since the Model Soup approach has been shown to have higher out-of-distribution performance than single models [31]. Additionally, it has yet to be demonstrated that the methods developed here will generalise between robotic laparoscopic surgery and traditional laparoscopic surgery. Given the high level of performance on zero-shot learning tasks [31], model soup trained on robotic laparoscopic tasks may perform well on traditional laparoscopic surgery classification and

evaluation with little to no additional fine-tuning.

References

[1] A. Johnson, Laparoscopic surgery, The Lancet 349 (9052) (1997) 631–635. doi:10.1016/S0140-6736(96)10032-5.

[2] B. Jönsson, N. Zethraeus, Costs and benefits of laparoscopic surgery—a review of the literature, European Journal of Surgery 166 (S12) (2000) 48–56.

[3] Y. Tanagho, et al., 2D versus 3D visualization: Impact on laparoscopic proficiency using the fundamentals of laparoscopic surgery skill set, Journal of Laparoendoscopic & Advanced Surgical Techniques 22 (9) (2012) 865–870. doi:10.1089/lap.2012.0220.

[4] S. Dreier, S. Mona, L. Konge, F. Bjerrum, Three-dimensional versus two-dimensional vision in laparoscopy: a systematic review, Surgical Endoscopy 30 (1) (2016) 11–23. doi:10.1007/s00464-015-4189-7.

[5] L. Villegas, et al., Laparoscopic skills training, Surgical Endoscopy And Interventional Techniques 17 (12) (2003) 1879–1888. doi:10.1007/s00464-003-8172-3.

[6] G. Azzie, et al., Development and validation of a pediatric laparoscopic surgery simulator, Journal of Pediatric Surgery 46 (5) (2011) 897–903. doi:https://doi.org/10.1016/j.jpedsurg.2011.02.026.

[7] E. Lukács, R. Levendovics, T. Haidegger, Enhancing autonomous skill assessment of robot-assisted minimally invasive surgery: A comprehensive analysis of global and gesture-level techniques applied on the JIGSAWS dataset, Acta Polytechnica Hungarica 20 (2023) 133–153. doi:10.12700/APH.20.8.2023.8.8.

[8] T. Haidegger, S. Speidel, D. Stoyanov, R. Satava, Robot-assisted minimally invasive surgery—surgical robotics in the data age, Proceedings of the IEEE 110 (2022) 835–846. doi:10.1109/JPROC.2022.3180350.

[9] C. D'Ettorre, A. Mariani, A. Stilli, F. Rodriguez y Baena, P. Valdastri, A. Deguet, P. Kazanzides, R. H. Taylor, G. S. Fischer, S. P. DiMaio,

A. Menciassi, D. Stoyanov, Accelerating surgical robotics research: A review of 10 years with the da vinci research kit, IEEE Robotics & Automation Magazine 28 (4) (2021) 56–78. doi:10.1109/MRA.2021.3101646.

[10] J. Emken, E. Mcdougall, R. Clayman, Training and assessment of laparoscopic skills, JSLS : Journal of the Society of Laparoendoscopic Surgeons 8 (2004) 195–9.

[11] M. Caban, et al., Use of collapsible box trainer as a module for resident education, JSLS 17 (3) (2013) 440–444. doi:10.4293/108680813X13693422521430.

[12] J. V. Taba, et al., The development of laparoscopic skills using virtual reality simulations: A systematic review, Plos one 16 (6) (2021) e0252609.

[13] M. Varras, et al., Role of laparoscopic simulators in the development and assessment of laparoscopic surgical skills in laparoscopic surgery and gynecology (review), World Academy of Sciences Journal 2 (2) (2020) 65–76. doi:10.3892/wasj.2020.41.

[14] Y. Gao, et al., JHU-ISI gesture and skill assessment working set (JIGSAWS): A surgical activity dataset for human motion modeling, Modeling and Monitoring of Comp. Assisted Interventions (2014).

[15] B. van Amsterdam, et al., Gesture recognition in robotic surgery: A review, IEEE Transactions on Biomedical Engineering 68 (6) (2021) 2021–2035. doi:10.1109/TBME.2021.3054828.

[16] N. Ahmidi, et al., A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery, IEEE Transactions on Biomedical Engineering 64 (9) (2017) 2025–2041. doi:10.1109/TBME.2016.2647680.

[17] A. Cotton, B. Sainsbury, C. Rossa, Sliding adaptive dynamic time warping for segment matching in surgical simulation data, in: Inter.Conf. on Smart Multimedia, 2024.

[18] K. MacWilliams, J. Green, A. Nasr, G. Azzie, C. Rossa, Optimizing sensor selection in laparoscopic simulators: Lessons learned in a robotic platform, in: 2025 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), IEEE, 2025, pp. 1–6.

[19] Y. Long, J. Y. Wu, B. Lu, Y. Jin, M. Unberath, Y.-H. Liu, P. A. Heng, Q. Dou, Relational Graph Learning on Visual and Kinematics Embeddings for Accurate Gesture Recognition in Robotic Surgery, arXiv preprint arXiv:2011.01619 v2, 29 Jun 2021 (Jun. 2021).
URL https://arxiv.org/abs/2011.01619

[20] C. Shi, Y. Zheng, A. Majewicz Fey, Recognition and Prediction of Surgical Gestures and Trajectories Using Transformer Models in Robot-Assisted Surgery, arXiv preprint arXiv:2212.01683Accepted at the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (Dec. 2022).

[21] D. Kiyasseh, R. Ma, T. F. Haque, B. J. Miles, C. Wagner, D. A. Donoho, A. Anandkumar, A. J. Hung, A vision transformer for decoding surgeon activity from surgical videos, Nature Biomedical Engineering 7 (2023) 780–796. doi:10.1038/s41551-023-01010-8.
URL https://doi.org/10.1038/s41551-023-01010-8

[22] S. Khalid, M. Goldenberg, T. Grantcharov, B. Taati, F. Rudzicz, Evaluation of Deep Learning Models for Identifying Surgical Actions and Measuring Performance, JAMA Network Open 3 (3) (2020) e201664. doi:10.1001/jamanetworkopen.2020.1664.

[23] L. Tao, L. Zappella, G. D. Hager, R. Vidal, Surgical gesture segmentation and recognition, in: Medical Image Computing and Computer-Assisted Intervention, Vol. 8151 of Lecture Notes in Computer Science, Springer, 2013, pp. 339–346. doi:10.1007/978-3-642-40760-4_43.

[24] S. Albasri, et al., A novel distance for automated surgical skill evaluation, in: 2019 7th E-Health and Bioengineering Conference, 2019. doi:10.1109/EHB47216.2019.8970029.

[25] J. A. Martin, G. Regehr, R. Reznick, H. MacRae, J. Murnaghan, C. Hutchison, M. Brown, Objective structured assessment of technical skill (OSATS) for surgical residents, British Journal of Surgery 84 (2) (1997) 273–278. doi:10.1046/j.1365-2168.1997.02502.x.

[26] E. Yanik, U. Kruger, X. Intes, R. Rahul, S. De, Video-based formative and summative assessment of surgical tasks using deep learning, Scientific

Reports 13 (1) (2023) 1038. doi:10.1038/s41598-022-26367-9.
URL https://doi.org/10.1038/s41598-022-26367-9

[27] J. Quarez, M. Modat, S. Ourselin, J. Shapey, A. Granados, ReCAP: Recursive Cross Attention Network for Pseudo-Label Generation in Robotic Surgical Skill Assessment, arXiv preprint arXiv:2407.05180Last revised 7 Jul 2025 (2024).
URL https://arxiv.org/abs/2407.05180

[28] D. Anastasiou, et al., Keep your eye on the best: Contrastive regression transformer for skill assessment in robotic surgery, IEEE Robotics and Automation Letters 8 (3) (2023) 1755–1762.

[29] Z. Li, L. Gu, W. Wang, R. Nakamura, Y. Sato, Surgical skill assessment via video semantic aggregation (08 2022). doi:10.48550/arXiv.2208.02611.

[30] A. Zia, I. Essa, Automated surgical skill assessment in rmis training, International Journal of Computer Assisted Radiology and Surgery 13 (5) (2018) 731–739. doi:10.1007/s11548-018-1735-5.
URL https://doi.org/10.1007/s11548-018-1735-5

[31] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, L. Schmidt, Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time, arXiv preprint arXiv:2203.05482Preprint (2022).
URL https://arxiv.org/abs/2203.05482

[32] E. Chaudary, S. A. Khan, W. Mumtaz, EEG-CNN-Souping: Interpretable emotion recognition from EEG signals using EEG-CNN-souping model and explainable AI, Computers and Electrical Engineering 123 (2025) 110189. doi:10.1016/j.compeleceng.2025.110189.

[33] R. C. Maron, et al., Model soups improve performance of dermoscopic skin cancer classifiers, European Journal of Cancer 173 (2022) 307–316. doi:10.1016/j.ejca.2022.07.002.

[34] T.-A.-Q. Nguyen, et al., Benign Paroxysmal Positional Vertigo Disorders Classification Using Eye Tracking Data, in: I. Maglogiannis, L. Iliadis,

J. Macintyre, M. Avlonitis, A. Papaleonidas (Eds.), Artificial Intelligence Applications and Innovations, Springer Nature Switzerland, Cham, 2024, pp. 174–185. doi:10.1007/978-3-031-63215-0_13.

[35] B. Neyshabur, H. Sedghi, C. Zhang, What is being transferred in transfer learning?, in: Advances in Neural Information Processing Systems, Vol. 33, 2020, pp. 512–523.
URL https://arxiv.org/abs/2008.11687

[36] N. Ahmidi, et al., A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery, IEEE Transactions on Biomedical Engineering 64 (9) (2017) 2025–2041. doi:10.1109/TBME.2016.2647680.

[37] Nguyen, et al., Benign paroxysmal positional vertigo disorders classification using eye tracking data, in: Artificial Intelligence Applications and Innovations, Springer Nature Switzerland, Cham, 2024, pp. 174–185.