PEg TRAnsfer Workflow recognition challenge report: Do multimodal data improve recognition?

A. Huaulme, K. Harada, Q.-M. Nguyen, B. Park, S Hong, M. K. Choi, M. Peven, Y. Li, Y Long, Q. Dou, et al.

Highlights

- Peg Transfer data set containing video, kinematic, semantic segmentation and workflow annotation.
- Challenge of surgical workflow recognition with different modality
- Comparison of multiple deep learning based recognition methods

PEg TRAnsfer Workflow recognition challenge report: Do multimodal data improve recognition?

Arnaud Huaulmé[a,*], Kanako Harada[b], Quang-Minh Nguyen[a], Bogyu Park[c], Seungbum Hong[c], Min-Kook Choi[c], Michael Peven[d], Yunshuang Li[e], Yonghao Long[f], Qi Dou[f], Satyadwyoom Kumar[g], Seenivasan Lalithkumar[h], Ren Hongliang[h,i], Hiroki Matsuzaki[j], Yuto Ishikawa[j], Yuriko Harai[j], Satoshi Kondo[k], Manoru Mitsuishi[b], Pierre Jannin[a,*]

[a] *Univ Rennes,INSERM, LTSI - UMR 1099, F35000, Rennes, France*
[b] *Department of Mechanical Engineering, the University of Tokyo,Tokyo 113-8656, Japan*
[c] *VisionAI hutom, Seoul, Republic of Korea*
[d] *Johns Hopkins University, Baltimore, USA*
[e] *Zhejiang University, Hangzhou, China*
[f] *Department of Computer Science & Engineering, The Chinese University of Hong Kong*
[g] *Netaji Subhas University of Technology, Delhi, India*
[h] *National University of Singapore, Singapore, Singapore.*
[i] *The Chinese University of Hong Kong, Hong Kong, Hong Kong*
[j] *National Cancer Center Japan East Hospital, Tokyo 104-0045, Japan*
[k] *Muroran Institute of Technology, Hokkaido, Japan*

Abstract

Background and Objective: In order to be context-aware, computer-assisted surgical systems require accurate, real-time automatic surgical workflow recognition. In the past several years, surgical video has been the most commonly-used modality for surgical workflow recognition. But with the democratization of robot-assisted surgery, new modalities, such as kinematics, are now accessible. Some previous methods use these new modalities as input for their models, but their added value has rarely been studied. This paper presents the design and results of the "PEg TRAnsfer Workflow recognition" (PETRAW) challenge with the objective of developing surgical workflow recognition methods based on one or more modalities and studying their added value.

Methods: The PETRAW challenge included a data set of 150 peg transfer sequences performed on a virtual simulator. This data set included videos,

*Corresponding author
Email addresses: arnaud.huaulme@univ-rennes1.fr (Arnaud Huaulmé), pierre.jannin@univ-rennes1.fr (Pierre Jannin)

kinematic data, semantic segmentation data, and annotations, which described the workflow at three levels of granularity: phase, step, and activity. Five tasks were proposed to the participants: three were related to the recognition at all granularities simultaneously using a single modality, and two addressed the recognition using multiple modalities. The mean application-dependent balanced accuracy (AD-Accuracy) was used as an evaluation metric to take into account class balance and is more clinically relevant than a frame-by-frame score.

Results: Seven teams participated in at least one task with four participating in every task. The best results were obtained by combining video and kinematic data (AD-Accuracy of between 93% and 90% for the four teams that participated in all tasks).

Conclusion: The improvement of surgical workflow recognition methods using multiple modalities compared with unimodal methods was significant for all teams. However, the longer execution time required for video/kinematic-based methods(compared to only kinematic-based methods) must be considered. Indeed, one must ask if it is wise to increase computing time by 2,000 to 20,000% only to increase accuracy by 3%. The PETRAW data set is publicly available at `www.synapse.org/PETRAW` to encourage further research in surgical workflow recognition.

Keywords: Surgical Process Model, Workflow recognition, Multimodal, OR of the future

1. Introduction

To fully integrate computer-assisted surgery systems in the operating room, a complete and explicit understanding of the surgical procedure is needed. A surgical process model (SPM) is a "simplified pattern of a surgical process that reflects a predefined subset of interest of the surgical process in a formal or semi-formal representation" [1], thus allowing for the surgical procedure to be rigorously modeled and described. The SPM methodology consists of decomposing a surgical procedure into five increasingly-coarse levels of granularity: dexeme,

surgeme, activity, step, and phase [2, 3]. A dexeme, the lowest granularity level, is a numeric representation of the motion. A surgeme represents a surgical motion with an explicit semantic interpretation of the immediate motion (e.g., pulling). An activity describes the motion's overall action (action verbs; e.g., cut) performed on a specific target (e.g., the pouch of Douglas) by a specific surgical instrument (e.g., a scalpel). A step is the succession of these activities which together achieve a specific surgical objective (e.g., resection of the pouch of Douglas). Finally, a phase is the succession of steps that constitute a main period of the intervention (e.g., resection). SPM's are used for learning and expertise assessment [4, 5], robot assistance [6], operating room optimization and management [7, 8], decision-making support [9], and quality supervision [10].

The primary limitation of the state-of-the-art in SPM's [3, 4, 5, 7, 9, 10] is their need to be manually interpreted by human observers, which is observer-dependent, time-consuming, and subject to error [11]. Thus, the proposed solutions can not be directly used to bring context-awareness into computer-assisted surgery applications in the operating room. To overcome this limitation, automatic workflow recognition methods have been developed for multiple granularity levels, including phase [8, 12, 13], step [14, 15], and activity [6, 16]. With the emergence of deep learning, most of these recent automatic workflow recognition methods are based on convolutional neural networks, such as AlexNet [17] or ResNet [18]; on recurrent neural networks, such as LSTM [19] or gated recurrent unit (GRU) [20]; and more recently on transformers [21].

Along with what methodology to use, it is also an open question as to which data modalities should be used as input for this task. In robot-assisted surgery and virtual reality training environments, video and kinematic data are both readily available. Despite this, most state-of-the-art workflow recognition methods are based on a single modality, such as only video [22, 23] or only kinematic data [3, 24]. Few studies have used workflow recognition method based on both video and kinematic data [25, 26, 27]. However, with the exception of the study by Long *et al.*[26], they do not compare the results obtained based on

the number and type of input modalities.

Semantic segmentation of surgical video is also essential for surgical understanding and is an active area of research. For example, in five editions of the EndoVis MICCAI Challenge (2015 to 2020), six of the 19 proposed sub-challenges were dedicated to this topic. However, to the best of our knowledge, semantic segmentation has rarely been used as a supplementary task paired with, or as additional input for, surgical workflow recognition.

Therefore, the "PEg TRAnsfer Workflow recognition by different modalities" (PETRAW) sub-challenge, which is part of EndoVis, provided a unique data set for automatic recognition of surgical workflows containing video, kinematic, and segmentation data on 150 peg transfer training sequences. Participants were asked to develop model(s) to recognize phases, steps, and activities using one or several of the available modalities.

2. Methods: Challenge Design

This section describes the challenge design, organization, objective, data set, and assessment methods.

2.1. Challenge organization

The PETRAW challenge was a one-time event organized as part of EndoVis during the online 2021 international conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI2021). Four people were involved in the organization: Arnaud Huaulmé and Pierre Jannin from the University of Rennes 1 (France), and Kanako Harada and Mamoru Misthuishi from Tokyo University (Japan). Complete information about the challenge was made available to participants using the Synapse platform: `www.synapse.org/PETRAW`.

Challenge participants were subject to the following rules:

- Participants had to submit a fully automatic method that could recognize phases, steps, and activities on the same model using one or several modalities; and

- Only data provided by the organizers and publicly available data sets, including pre-trained networks, were authorized for use in training. The publicly available data sets must have been open or otherwise available to all participants at the time the PETRAW data set was released.

The results of all participating teams were announced publicly during the challenge day. Challenge organizers and people from the organizing institutions could also participate in the challenge but were excluded from the competitive rankings. Participating teams were encouraged (but not required) to provide their code as open access.

For a valid submission, the participating teams had to provide the following elements: a write-up, a Docker image allowing the organizers to compute the results, and a pre-recorded talk to limit technical issues during the challenge day (online event). Multiple Docker images could be submitted, but only the last submission was officially used to generate the evaluation results. No leaderboard or evaluation results were provided prior to the challenge day.

The challenge schedule was as follows: The training data set, including videos, kinematic data, and workflow annotations, was released on June 1, 2021; corresponding semantic segmentation data was released on June 9, 2021; submissions were accepted until September 12, 2021 (23:59 PST); and the evaluation results were announced on October 1, 2021, during the online MICCAI2021 event. Some teams obtained unexpectedly poor results (i.e., workflow recognition rates inferior to 50%), which made further analysis of the results not relevant. Therefore, each team was allowed to provide a new submission before October 31, 2021. The teams that made a new submission are identified in Section 3.2. The challenge test data set and the organizers' evaluation scripts were released with this paper at `www.synapse.org/PETRAW`

2.2. Challenge objective

The objective of the PETRAW challenge was to study the contribution of each modality (either alone or in combination) to surgical workflow recognition. To achieve this goal, participants were asked to create a single classification model

to determine the surgical task at three levels of granularity (phase, step, and action). Five different tasks were offered as part of the challenge: three concerned the development of unimodal models (i.e., video-based, kinematic-based, or semantic segmentation-based models); and two concerned multimodal-based models. The unimodal-based models were used as a baseline for comparison with the multimodal-based models. In order to keep to a reasonable number of tasks, not all multimodal configurations could be studied. For models based on semantic segmentation data (and to reflect the fact that clinically this modality can be only obtained through a trained segmentation model), participants were asked to use the output of such model as input for PETRAW.

2.3. Challenge data set

The challenge data set was composed of 150 sequences of peg transfer training sessions. The objective of the peg transfer session was to transfer six blocks from the left peg to the right and then back. Each block needed to be extracted from the peg using a grasper (operated by one hand), transferred to the other grasper (in the other hand), and finally inserted onto the peg on the opposite side of the board.

All sequences were acquired by a non-medical expert at the LTSI Laboratory, University of Rennes 1, France. The data set was divided into training data (n=90 sequences) and test data (n=60 sequences). Each sequence included kinematic data, video, semantic segmentation of the video for each frame, and workflow annotations at each level of granularity. Only the training data set was provided to participants.

2.3.1. Data acquisition

The challenge data was acquired on a virtual reality simulator (Figure 1) developed at the Department of Mechanical Engineering, University of Tokyo, Japan [28], consisting of a laptop (i7-700HQ, 16Go RAM, GTX 1070), a 3D rendering setup (3D screen: 24 inches, 144Hz; and 3D glasses), and two haptic user interfaces (3D system Touch^{TM}).

Figure 1: The virtual reality simulator used for data acquisition.

For data acquisition, a single operator performed a series of five consecutive peg transfer tasks followed by a break of at least 5 hours to limit fatigue. This was repeated 30 times to yield a total of 150 peg transfer task sequences. The COVID-19 crisis (acquisition made in 2020-2021) did not allow us to recruit multiple participants. To limit the effect of immediate learning or fatigue in a single session, three sequences from each series were randomly chosen for training, and the remaining two for testing.

The kinematic data and videos were synchronously acquired at 30 Hz during each peg transfer task. Each video had a resolution of 1920x1080 pixels and semantic segmentation was performed for each frame off-line following the task. Kinematic data included the position, rotation quaternion, forceps aperture angle, linear velocity (obtained from simulation, not derived from position), and angular velocity (obtained from simulation, not derived from orientation) of the left and right instruments (i.e., graspers). The position and linear velocity were measured in centimeters and centimeters per second, respectively. The angle and angular velocity were measured in degrees and degrees per second, respectively.

The semantic segmentation included six classes (shown in Figure 2): back-

ground (black, hexadecimal code:#000000), base (white, #FFFFFF), left instrument (red, #FF0000), right instrument (green, #00FF00), pegs (blue, #0000FF), and blocks (magenta, #FF00FF).

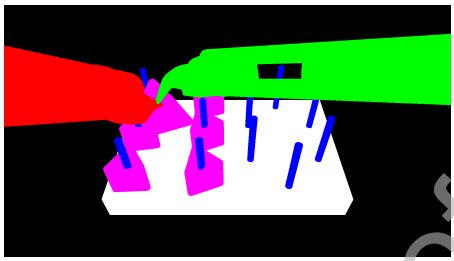


Figure 2: Representative segmentation mask with the six classes: background (black), base (white), left instrument (red), right instrument (green), pegs (blue) and blocks (magenta).

The workflow annotations were automatically computed using the scene information and the ASURA method [11]. The challenge organizers had previously demonstrated in [11] that ASURA is more accurate and robust than manual annotation on peg transfer tasks. Two phases, twelve steps, six action verbs, two targets, and one surgical instrument were identified to describe the workflow (Table 1). Each phase corresponded to the transfer of all of the blocks in one direction (e.g. "L2R" for left to right). Each step (six per phase) corresponded to the transfer of a single block (e.g."Block1 L2R" for the transfer of the first block from the left to the right). For the activities, two targets were differentiated: "block" and "other block". "Block" corresponds to the one that is currently being transferred. "Other block" is an additional target used to differentiate when the user accidentally interacts with any block other than the one to be transferred.

One limitation of the method presented by [11] was the inability to accurately differentiate between the action verbs "catch" and "touch", as each tool tip was considered as a unique virtual object. The virtual reality simulator was updated to include four separating regions rather than one, allowing these actions to be readily differentiated. Accordingly, the workflow annotations were manually examined and corrected to ensure annotation quality.

Table 1: Peg-transfer vocabulary.

Phases	Steps
Transfer Left To Right (L2R)	Block 1 L2R
	Block 2 L2R
	Block 3 L2R
	Block 4 L2R
	Block 5 L2R
	Block 6 L2R
Transfer Right To Left (R2L)	Block 1 R2L
	Block 2 R2L
	Block 3 R2L
	Block 4 R2L
	Block 5 R2L
	Block 6 R2L

Activities		
Verb	Target	Tool
Catch	Block	Grasper
Drop	Other block	
Extract		
Hold		
Insert		
Touch		

2.3.2. Data pre-processing

The original workflow annotations were formatted in terms of start and finish time, expressed in milliseconds. These annotations were sampled to provide a discrete sequence at 30Hz, synchronized with the kinematic, video, and segmentation data to allow for frame-by-frame annotation. Due to their lack of variability, the two targets and the tool were not included in the workflow annotation. Furthermore, when no phase, step, or activity occurred, the term "idle" was used. For each timestamp, the following information was provided: timestamp_number, phase_value, step_value, verb_Left_Hand, verb_Right_Hand.

2.3.3. Ground truth uncertainties

The semantic segmentations were the primary source of uncertainty in the ground truth. Due to the transformation of 3D meshes into 2D images, some pixels were attributed to the wrong class, especially at boundaries between the

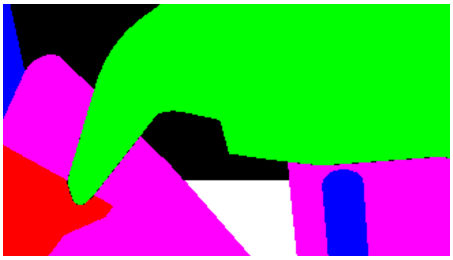


Figure 3: Zoom of 219x123 pixels from Figure 2 to highlight segmentation errors. Right instrument/block (green/magenta) and left/right instruments (red/green) errors are shown where pixels are labeled as background (black). On this zoom, only 51 pixels were miss-segmented (around 0.2%).

right instrument/peg, left instrument/peg, left instrument/block, and left/right instruments (Figure 3). We estimated this uncertainty by counting the number of mis-segmented pixels on 10 images that included many boundary regions, such as those between surgical instruments, pegs, and blocks. On each image, the number of mis-segmented pixels represents less than 0.25% of the total image. To take into account the fact that this manual assessment was not representative of the whole data set, we estimated that this mis-segmentation represents less than 0.5% of pixels.

Workflow annotations were another source of uncertainty. Although the ASURA method is consistent (i.e., it generates the same result in two identical situations) and a manual check was performed to limit inaccuracies, some components could not be recognized with complete certainty. Two particular instances were identified. First, in sequence 130 of the training data set, the block in step “Block 1 R2L” was inserted in a non-standard way. Specifically, the block was released by the operator, and while falling became inserted in the peg. Therefore, the insert action was absent. The other instance concerned sequence 79 of the test data set. This time, the operator caught a block before the previous one had been fully inserted, leading to an overlap between the steps “Block 5 R2L” and “Block 6 R2L”. The second was chosen as the sole annotation to maintain the true beginning of the step.

2.3.4. Data set characteristics

The training and test data sets presented similar characteristics. The mean and standard deviation duration was 140.2 ± 18.9 seconds for the training data set and 141.7 ± 18.0 seconds for the test data set. Figure 4 presents the distribution of every vocabulary component for each granularity level in the training data set (Figures 4a, 4c, 4e, 4g) and the test data set (Figures 4b, 4d, 4f, 4h). Even for underrepresented components, the distribution was very similar in both data sets. For instance, the verb "touch" (left hand) represented 0.59% and 0.60% of the samples in the training and test data sets, respectively, and "touch" (right hand) represented 0.62% and 0.48%, respectively. The distribution of each vocabulary component between each data set is only statistically different (Mann-Whitney test) for two steps: "Block 1 L2R" and "Block 6 L2R", with p=0.045 and p=0.036 respectively.

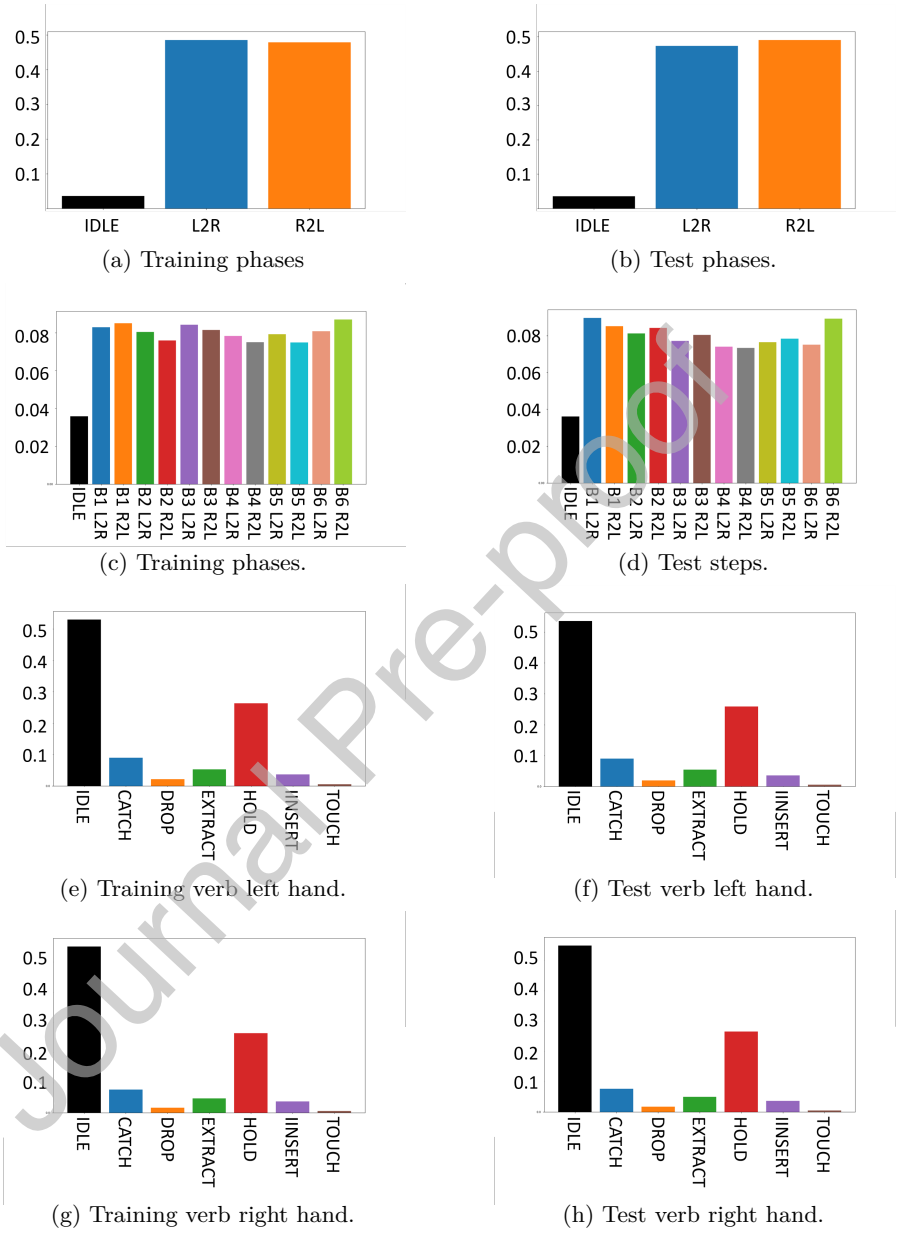
Another important characteristic of the data sets was the high class unbalance of at least one vocabulary term for each granularity level. For the phases, the term "idle" represented less than 4% of all data, whereas the other phase terms accounted for more than 47% (L2R and R2L). For the steps, the term "idle" represented less than 4%, whereas the non-idle steps accounted for approximately more than 7.5% of each data set(Figures 4a-4d). This unbalance was more pronounced at the action level, where the least represented verb (i.e., "touch") represented approximately 0.6% of the data set, whereas the verb "idle" accounted for more than 53%. The detailed distribution values for each granularity level in both data sets are provided in supplementary material.

2.4. Assessment method

2.4.1. Metrics

To assess the participants' workflow recognition models and to take into account the high class unbalance, balanced versions of accuracy, precision, recall, and F1 were used.

In practice, however, some small variations in surgical task recognition are not clinically meaningful and do not constitute a true error. Motivated by this,



(a) Training phases

(b) Test phases.

(c) Training phases.

(d) Test steps.

(e) Training verb left hand.

(f) Test verb left hand.

(g) Training verb right hand.

(h) Test verb right hand.

Figure 4: Distribution of each term at each granularity level in the training and test data sets.The y-axis represents the percentage of frames. In (a) and (b), "L2R" means transfer left to right and "R2L" means transfer right to left. In (c) and (d), "B1 L2R" means block 1 left to right, "B2 L2R" means block 2 left to right.

Dergachyova *et al.* [29] proposed a re-estimation of these classic frame-by-frame scores, called application-dependent scores, to take into account an acceptable delay d. When a predicted transition occurs within a transition window ($2d$) centered on the ground truth transition, all frames between the two transitions are considered correct if it is the same transition type (e.g. transition for verb "catch" or verb "extract"). Therefore, the balanced application-dependent accuracy (AD-Accuracy) was used and the acceptable delay was fixed at 250 ms.

To assess the participants' segmentation models, the mean Intersection-Over-Union (IoU) over all classes was also used, also known as the Mean Jaccard Index over all classes. The IoU is the area of overlap between the predicted segmentation ($Pred$) and the ground truth (GT), divided by the area of union between the $Pred$ and the GT. In our cases, there was a multi-class segmentation problem, therefore the mean IoU value of the image was calculated by taking the IoU of each class and averaging it over the classes:

$$\begin{aligned} MeanIoU_{frame} &= \frac{1}{6}\sum_{class} IoU_{class} \\ &= \frac{1}{6}\sum_{class} \frac{|GT \cap Pred|_{class}}{|GT \cup Pred|_{class}} \\ &= \frac{1}{6}\sum_{class} \frac{TP_{class}}{TP_{class} + FP_{class} + FN_{class}}, \end{aligned} \tag{1}$$

where TP (True Positives) is the number of pixels inside the GT area that are correctly predicted, FP (False Positives) is the number of pixels outside the GT area but predicted as belonging to the class, and FN (False Negatives) is the number of pixels inside the GT area that are incorrectly predicted.

2.4.2. Ranking method

The ranking of the participating methods used only the surgical task recognition metrics. Metrics computed for evaluating the segmentation models were provided for information purposes only.

A metric-based aggregation method using the AD-Accuracy values across all test sequences was used for the ranking. Metric-based aggregation was used

according to the recommendations made in [30], which show it to be one of the most robust. As all tasks consisted of recognizing the phase, step, and the actions of the left and right hands (i.e., the left and right verbs), the ranking score for the algorithm a_i was computed as follows:

$$s(a_i) = \frac{s_{phase}(a_i) + s_{step}(a_i) + s_{verb\_left}(a_i) + s_{verb\_right}(a_i)}{4} \tag{2}$$

with,

$$s_{phase}(a_i) = \frac{\sum_{t=0}^{T} phase\_balance\_accuracy\_case\_t}{T}, \tag{3}$$

where T is the number of sequences to test. Similar equations were used for the other terms ($s_{step}(a_i)$, $s_{verb\_left}(a_i)$ and $s_{verb\_right}(a_i)$) with a numerator specific to each, i.e., $\sum_{t=0}^{T} step\_balance\_accuracy\_case\_t$ for $s_{step}(a_i)$, etc.

If a participant method did not produce a prediction for one or several granularity levels, the accuracy given for each missing granularity level was that expected for uniformly random predictions. For example, if a model did not predict the phase, s_{phase} would be set to 1/3 corresponding to the phase having 3 potential values. In practice, this was not encountered and each evaluated model produced results for each level of granularity.

Ranking stability was assessed by testing different ranking methods: meanThenRank, medianThenRank, rankThenMean, rankThenMedian, and testBased. MeanThenRank was chosen for the ranking. MedianThenRank differs from the previous method because it used the median instead of the mean in equation 3. For rankThenMean and rankThenMedian, first, the results of each sequence were ranked among participants, and then the final results were the mean or median of all ranks. The testBased method is based on bootstrapping. The ranking was considered stable if a team was ranked in the same position with the majority of ranking methods. If the ranking was not stable according to the chosen methods, a tie between teams was pronounced. The ranking computation and analysis were performed with the ChallengeR package provided by [31].

2.4.3. Online recognition compatibility

To be online compatible, the proposed methods must satisfy two conditions:

- to produce predictions faster than the duration between the two samples (i.e., faster than 30 Hz); and
- to be causal (i.e., not use data from a future time point to make predictions).

The computation time was not studied because it could not be assessed fairly for all teams. Indeed, the teams provided a unique Docker image for all tasks, and some teams did not write the output file to standard output as it was received, which did not allow for their durations to be precisely measured.

To verify that the methods were causal, the online availability of the frames was mimicked. One additional sequence of 10 seconds, corresponding to the transfer of the first block from the left to the right, was recorded. This sequence was used to generate 300 sub-sequences, each one a frame longer than the previous. Thus, the first sequence only contained the information of the first frame, the second one contained the information of the two first frames, etc. The models were run on the 300 sub-sequences and the last prediction of each sub-sequence to create a definitely causal prediction sequence. A method was considered causal if and only if this definitely causal prediction sequence was identical to the prediction sequence given by the full 300 frames. This causality-testing method is fully automated and also takes into account the complete pipeline used to perform the prediction, such as pre- and post-processing steps, which could lead to a non-causal method even if the network only uses causal components. For reasons of computation time and environmental responsibility, this test was not performed on a whole sequence or the whole test data set. By testing the entire data set, we could be more confident in the causality of the proposed methods, but this would quickly display diminishing returns.

2.5. Additional analyses

To further analyze the impact of using multimodal instead of unimodal models, we performed two additional analyses that were not initially included in the challenge design: the statistical significance to use multimodal models instead of unimodal models, and the execution time. These additional analyses

only concerned the teams that participated in the multimodal tasks (4 and 5) with a combination of the same or similar models used for the unimodal tasks.

2.5.1. Comparison between unimodal and multimodal models

To assess the impact of each modality and its combinations on automatic workflow recognition, we performed a statistical analysis with the Wilcoxon test. The difference was significant if the p-value was inferior to 0.05.

2.5.2. Execution time

Performance is not the only important factor when developing automatic recognition models. Indeed, environmental aspects must also be taken into account [32]. To answer this question, we examined the execution time to compute the results of the 60 test sequences. These durations were interpolations that assumed the predictions in each task were computed independently and not the real execution time. Indeed, one team (Hutom, see section 3.2.1) used the predictions from tasks 1 to 3 as input for those of tasks 4 and 5, so the interpolation for the multimodal tasks took into account the execution time for the unimodal ones.

3. Results: Reporting of the Challenge Outcomes

3.1. Challenge submission

By September 12, 2021, 29 participants had registered for the PETRAW challenge: 17 were members of one of the six competing teams. The organizers also submitted results as a non-competing team to provide a baseline. As explained in Section 2.1, some teams obtained unexpected results and three teams resubmitted results for at least one task.

3.2. Information on the participating teams and their methods

This section describes each team, the methods they used, and the tasks in which they participated. Competing teams are presented in alphabetical order and not in terms of their ranking.

3.2.1. Hutom

The Hutom team (Bogyu Park, Seungbum Hong, and Minkook Choi from VisionAI hutom) participated in all proposed tasks. They resubmitted a Docker image for all tasks except the kinematic-based recognition task.

Before training, they performed a simple pre-processing step. To preserve temporal information, they split data into clips of 8 frames. They normalized kinematic data by standardizing the raw input without data augmentation. They resized video data to 256×256 pixels, followed by random cropping (224×224 pixels) and normalization. The cropping was limited to preserve the spatial information in each frame of the clip. They resized segmentation data to 512×512 pixels.

They used a similar baseline architecture for tasks based on the same modality. They computed segmentation data from the video recording using a DeepLabV3+ architecture [33]. They used a 3D ResNet network [34] for workflow recognition based on the video modality. For the segmentation modality, they used a SlowFast50 network [35] for segmentation-based recognition and a 3D ResNet network for video/kinematic/segmentation-based workflow recognition. They inputted kinematic data on a bi-directional long short-term memory (Bi-LSTM) network [36]. For multimodal recognition tasks, they used a convolutional feature fusion layer to efficiently perform the fusion of the feature output of each modality. They obtained embedding features with individual modal inputs from each model trained accordingly. Then, they compared the embedding features of each modality with those of other modalities to learn the different representations of each modality. They used the stop gradient-based SimSiam method [37] to compare representations between embedding features. Concomitantly, they stacked embedding features by modality into one block as a chunk and fused them into one embedding through a convolution operation. The approach assumed that feature elements for each modality in the same column have similar temporal information in similar positions. For all networks, they used the Adam optimizer and an initial learning rate of $1e^{-3}$, with a combination of Equalization

loss v2 [38] and Normsoftmax Loss [39] as long-tail recognition for addressing data imbalance.

3.2.2. JHU-CIRL

The JHU-CIRL team (Michael Peven and Gregory D. Hager; Johns Hopkins University) participated in the kinematic-based workflow recognition task.

They performed an under-sampling of the kinematic data to reduce the time dimension size in order to prevent vanishing gradient issues during training. For the test, they used the same under-sampling. The JHU-CIRL team did not perform any other pre-processing because they considered that besides the positional data, the addition of velocity data was sufficient for the recognition.

They used a unidirectional LSTM network [40] to recognize the four workflow components. They trained the model using traditional cross-entropy loss and the Adam optimizer. They paid special attention to the selection of the following hyperparameters: sampling rate, learning rate, LSTM hidden dimension size, and the number of layers in the LSTM. They ran 5-fold cross-validation to obtain results from each of these hyperparameters. Then, they selected the best set of hyperparameters for the final training: 15Hz sampling rate, $1e^{-3}$ learning rate, 256 LSTM Hidden dimension, and 2 LSTM layers.

3.2.3. MedAIR

The MedAIR team (Yunshuang Li, Yonghao Long, and Qi Dou, Zhejiang University and the Chinese University of Hong Kong) participated in three tasks: video-based, kinematic-based, and video/kinematic-based workflow recognition. They resubmitted a Docker image for the video-based workflow recognition task.

The MedAIR team resized videos to 224×224 pixels and then augmented the data using a random horizontal flip and a random rotation of $5°$. For kinematic data, they used a linear layer to obtain 2048 dimensions from the 28 dimensions to enrich the information.

For unimodal-based workflow recognition (video-based and kinematic-based tasks), the MedAIR team used a Trans-SVNet model [41]. First, they trained

two different convolutional neural networks (CNN) to extract spatial features, one for steps and another for left and right verbs. Then, they trained three multi-stage temporal convolutional networks (TCN) to obtain temporal features for steps and verbs. Finally, they used three transformer layers to combine spatial and temporal features to obtain the final output for the three labels. Phases were not directly predicted by the networks, but identified based on the predicted step. They used a stochastic gradient descent (SGD) optimizer with a cross-entropy loss and a learning rate of $5e^{-4}$.

For multimodal-based workflow recognition (video/kinematic-based task), they used a multi-modal relational graph network (MRG-Net) [26]. Like for unimodal-based workflow recognition, they used two CNNs to extract features from each frame in the video for steps and verbs. Then, they obtained the step labels using the original MRG-Net structure, which was the result of the fully connected layer with the output of three nodes in the graph. For the verb labels, the MedAIR team used fully connected layers to produce outputs k_t^l and k_t^r, the final label prediction for left and right verb labels. They identified phases based on the predicted step. They used an Adam optimizer with cross-entropy loss and learning rate of $1e^{-4}$.

3.2.4. MMLAB

The MMLAB team was composed of Satyadwyoom Kumar, Lalithkumar Seenivasan, and Hongliang Ren from the Netaji Subhas University of Technology, National University of Singapore, and the Chinese University of Hong Kong. They participated in the video/kinematic-based recognition task.

MMLAB team proposed a multi-task learning model to perform the recognition. First, each video frame was resized to 224×224 pixels. A ResNet 50 [18] pre-trained on ImageNet was used to extract visual features for each video frame. These features were passed with the frame-specific kinematic data through four label-specific networks (one per component). Each label-specific network was composed of two LSTMs [19], one for each modality, to capture the temporal features. The sequential length was set to 5, allowing the model to infer based

on the current and past 4 temporal information sets. The resulting temporal features were then passed through a single linear layer for recognition. Each label-specific network was trained independently with cross-entropy loss, Adam optimizer, and a learning rate of $1e^{-3}$ for phase and step recognition, and $1e^{-2}$ for hand verbs.

3.2.5. NCC NEXT

The NCC NEXT team (Hiroki Matsuzaki, Yuto Ishikawa, Kazuyuki Hayashi, Yuriko Harai, and Nobuyoshi Takeshita, National Cancer Center Japan East Hospital) participated in all proposed tasks. They resubmitted a Docker image for all tasks except the kinematic-based recognition task.

They resized the initial video frames to a resolution of 512×256 pixels for video-based workflow recognition and of 480×270 pixels for segmentation-based workflow recognition. This was followed by normalization. They did not perform any preprocessing of kinematic data.

For video-based workflow recognition they used Xception networks [42] pre-trained on ImageNet, one per component. They used the Radam optimizer [43] with different learning rates with a batch size of 4, $1e^{-3}$ for phases and steps, and $1e^{-4}$ with a cosine decay scheduler for hand verbs. They also used cross-entropy loss.

For kinematic-based workflow recognition, the NCC NEXT used the light gradient boosting machine (LightGBM) framework [44]. Like for the previous task, they did the training and tuning of hyperparameters (i.e., learning rate, minimum data in leaf, number of iterations, and number of leaves) separately for each component (Table 2). They chose gradient boosting as a predictor optimizer and the mean absolute error (MAE) as loss of function.

Parameters	Phase	Step	Verb_Left	Verb_Right
Learning rate	0.1	0.05	0.05	0.05
min_data_in leaf	9	9	3	9
num_iteration	200	100	100	50
num_leaves	11	31	11	11

Table 2: Hyperparameters for the kinematic based model developed by the NCC NEXT team

The segmentation was performed by a Deeplabv3+ architecture [33] with an Xception backbone pre-trained on the Pascal visual object classes (PascalVOC) data set [45]. With the predicted segmentation, they trained a multi-output classification model, based on the EfficientNetB7 architecture [46], with Radam optimizer, cross-entropy loss function, a learning rate of 0.0001 with a cosine decay scheduler, and a batch size of 16.

For the multimodal workflow recognition tasks, the NCC NEXT team selected the method used in the three previous tasks that displayed the highest accuracy for each component. Specifically, for video/kinematic-based workflow recognition task, they used the video-based architecture for phase and step recognition and the kinematic-based architecture for hand verb recognition. For the video/kinematic/segmentation-based model, they used the video-based architecture for phase recognition, the segmentation-based architecture for step recognition, and the kinematic-based architecture for hand verb recognition.

3.2.6. SK

The SK team (Satoshi Kondo, Muroran Institute of Technology) participated in all proposed tasks.

For preprocessing, the SK team resized the images to 640×353 pixels and then used random shifting (maximum shift size of 10% of the image size), scaling (0.9 to 1.1 times), rotation (-5 to 5 degrees), color jitter (-0.9 to 1.1 times for brightness, contrast, saturation, and hue), and Gaussian blurring (maximum sigma value = 1.0) for data augmentation. Finally, the images were normalized

and the kinematic data were normalized in each dimension.

For the video-based workflow recognition task, the SK team used an 18-layer ResNet network [18], pre-trained on ImageNet. The SK team omitted the final fully-connected layer of ResNet and fed its input 512-dimensional feature vector into two fully-connected layers to obtain a prediction of the step and hand verbs. Between these fully-connected layers, they inserted one ReLU and Dropout layers. The team used an Adam optimizer, with learning rate changes with cosine annealing with an initial value of $7.2e^{-4}$, and a batch size of 96. The team optimized the initial learning rates for each task with the Optuna library [47]. The team chose cross-entropy loss as the loss function, with weights for each class depending on the class frequency for hand verbs. Phases were not directly predicted from the image, but identified based on the predicted step.

The SK team used a stacked LSTM [19] with two layers and 28 hidden layers for the kinematic-based workflow recognition task. The LSTM output was fed into three fully connected layers as done for the previous task. The same optimizer and loss function were used. The initial learning rate was $1.5e^{-3}$ with a batch size of 6 and the number of data in a sequence was 30.

Image segmentation was done using the U-Net architecture [48] with ResNet18 as encoder with the summation of cross-entropy loss and dice loss. The SK team exploited the same model used for the video-based workflow recognition task and for the segmentation-based task. Both models were trained separately with an Adam optimizer and an initial learning rate of $2.4e^{-5}$ with a batch size of 32 for segmentation, and a learning rate of $1e^{-4}$ with a batch size of 6 for recognition.

For the video/kinematic-based task and video/kinematic/segmentation-based task, the SK team ensembled the previously trained dedicated modality networks to obtain a new prediction. As the SK team used the network parameters trained for the previous task, they did not train any network for these tasks.

3.2.7. MediCIS: non-competing team

The MediCIS team was a non-competing team due to the presence of challenge organizers (Quang-Minh Nguyen and Arnaud Huaulmé, University of Rennes 1).

The team participated in all proposed tasks.

For the preprocessing step, they resized the frames to 256×512 pixels. Additionally, to train the segmentation model, they down-sampled the data to 6 Hz. They z-normalized the kinematic data.

For the video-based workflow recognition task, the MediCIS team used a hierarchical RestNet50 network [18] pre-trained on ImageNet to extract spatial features. Then, they used a Multi-Stage Temporal Convolutional Network called MS-TCN++ [49], with two stages, trained from scratch.

For the kinematic-based workflow recognition task, they directly used data as features for a two-stage MS-TCN++.

They selected as their segmentation model a U-Net [50] network trained from scratch with the Adam optimizer, cross-entropy loss, learning rate of $1e^{-4}$, and batch size of 10. Like for the video-based task, workflow recognition was done by hierarchical ResNet50 followed by a two-stage MS-TCN++.

For the video/kinematic-based and video/kinematic/segmentation-based tasks, the MediCIS team extracted unimodal spatial features using a hierarchical ResNet50 network for video and segmentation data, followed by concatenation. Then, they trained a two-stage MS-TCN++.

They trained all workflow recognition models with the Adam optimizer, cross-entropy loss, learning rate of $1e^{-4}$, and batch size of 2. For the hierarchical ResNet50 network, they emphasized the training for granularities that are harder to recognize using the following weights in the loss: 1 for phases, 2 for steps, and 5 for both action verbs. They set the number of dilated convolutional layers in MS-TCN++ to 10, except for the first layer where it was 11. The number of feature maps for each layer was 64.

3.3. Workflow recognition results

All results were computed on the organizers' hardware via the provided Docker images. This section only presents the results used for the ranking (balanced AD-Accuracy). Other results, such as application-dependent scores for each

sequence and task, for each participating team, are available as supplementary material and at `www.synapse.org/PETRAW`.

3.3.1. Task 1: Video-based workflow recognition

Task 1 consisted of recognizing phases, steps, and hand verbs using video data only. Table 3 summarizes the algorithms used by the five teams that submitted models for this task.

Team	Hutom *	MedAIR *	NCC Next *	SK	MediCIS
Preprocessing	X	X	X	X	X
Augmentation	X	X		X	
Model	3DResNet	Trans-SVNet	Xception	ResNet18	ResNet50 & MS-TCN++
Optimizer	Adam	SGD	Radam	Adam	Adam
Loss	Equalization v2 & Normsoftmax	cross-entropy	cross-entropy	cross-entropy	cross-entropy
Learning Rate	$1e^{-3}$	$5e^{-4}$	$1e^{-3}$ & $1e^{-4}$	$7.2e^{-4}$	$1e^{-4}$
Causal					X

Table 3: Algorithms used for task 1. Teams that resubmitted models are highlighted with an asterisk. An "X" means that the method performed preprocessing, data augmentation, or is causal.

Comparison of the mean AD accuracy values for each test sequence (all models) (Figure 5) showed only a slight performance decrease (from 95.1% to 82.2%), but sequences 79 and 54 displayed the lowest performance (77.7% and 72.9%, respectively). Moreover, for all the test sequences, one model displayed lower AD-Accuracy values than the other models.

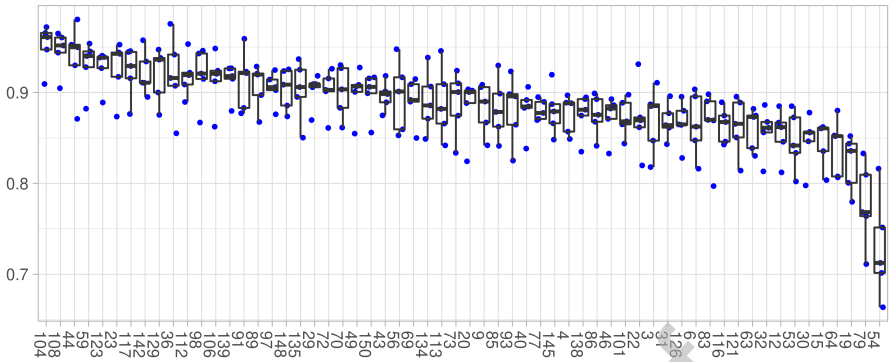


Figure 5: Task 1 recognition AD-Accuracy values (%) for each sequence. Each dot represents the AD-Accuracy of one model. The x-axis represent the test sequence id.

Comparison of the mean AD-Accuracy value for each model (Figure 6 showed that team SK and team Hutom, obtained the highest values (>90%), followed by team MediCIS and team NCC NEXT (>87%). MedAIR obtained the lowest results (≈84%).

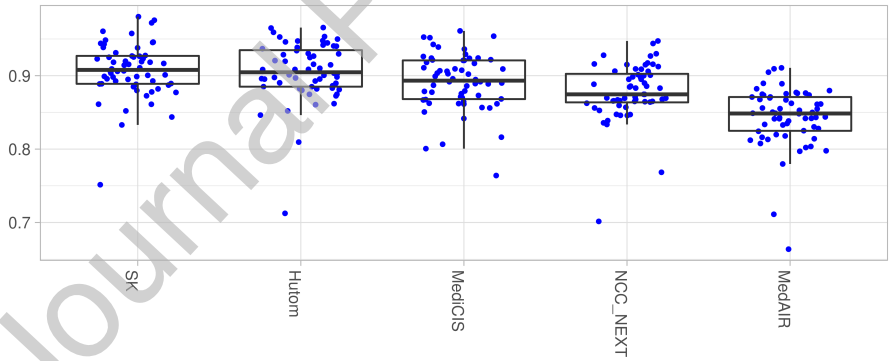


Figure 6: Mean task 1 recognition AD-Accuracy for each model. Each dot represents the AD-Accuracy for one sequence.

Team ranking was not influenced by the chosen method (Figure 7), except for the ranking of the SK and Hutom teams using the rankThenMedian and testBased methods.

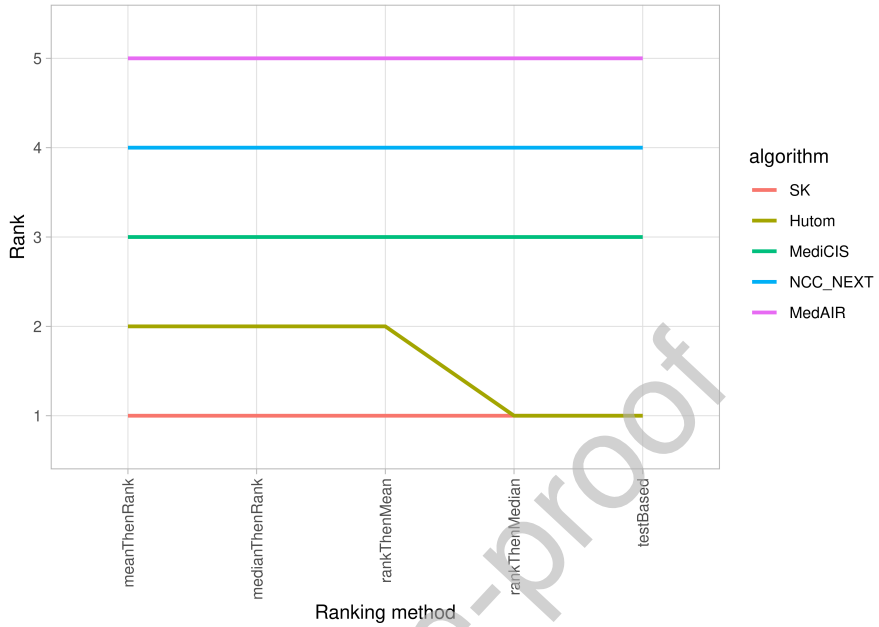


Figure 7: Task 1 recognition ranking stability using different ranking methods. Rank 1 indicates the best method.

3.3.2. Task 2: Kinematic-based workflow recognition

Task 2 consisted of recognizing phases, steps, and hand verbs using kinematic data only. Table 4 summarizes the methods used by the six participating teams for this task.

As with task 1, the performance per sequence slightly decreased (Figure 8). The highest AD-Accuracy values were superior to 90% for all teams. Three sequences (including sequences 79 and 54) had mean AD-Accuracy values inferior to 80%. Unlike task 1, the majority of sequences did not have outliers.

Team	Hutom	JHU-CIRL	MedAIR	NCC Next	SK	MediCIS
Preprocessing	X	X	X	X	X	X
Augmentation						
Model	Bi-LSTM	Uni-LSTM	Trans -SVNet	LightGBM	Stacked -LSTM	MS-TCN++
Optimizer	Adam	Adam	SGD	Gradient Boosting	Adam	Adam
Loss	Equalization v2 & Normsoftmax	cross-entropy	cross-entropy	MAE	cross-entropy	cross-entropy
Learning Rate	$1e^{-3}$	$1e^{-3}$	$5e^{-4}$	$1e^{-1}$ & $5e^{-2}$	$1.5e^{-3}$	$1e^{-4}$
Causal		X		X	X	

Table 4: Summary of the models used for task 2. An "X" means that the method performed preprocessing, data augmentation, or is causal.

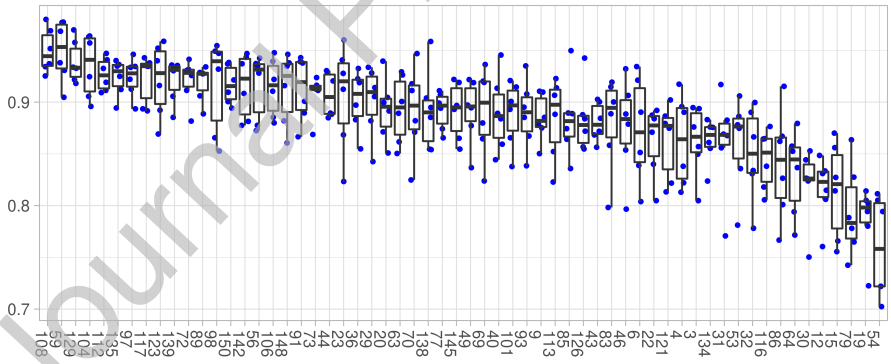


Figure 8: Task 2 recognition AD-Accuracy for each sequence. Each dot represents the AD-Accuracy of one model.

Results were very similar among teams (Figure 9). Four had a mean AD-Accuracy value of between 89.7% and 90.7%, and the other two displayed mean AD-accuracy values of 86.4% and 84.3%, respectively.

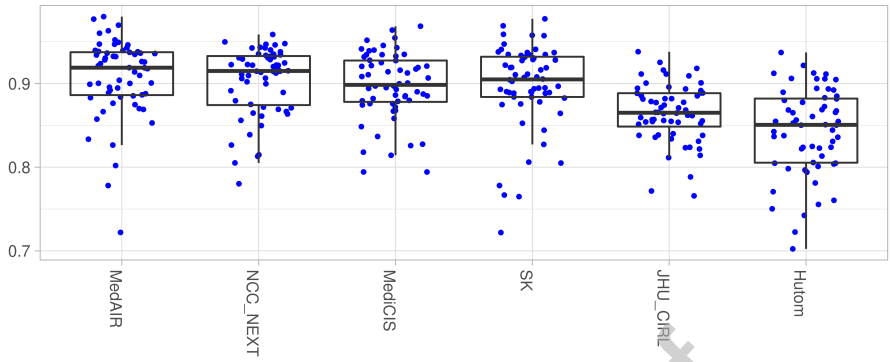


Figure 9: Mean task 2 recognition AD-Accuracy for each model. Each dot represents the AD-Accuracy for one sequence.

559 Ranking was not stable for team SK and team MediCIS (Figure 10). As
560 MediCIS was a non-competing team, SK was ranked third for this task.

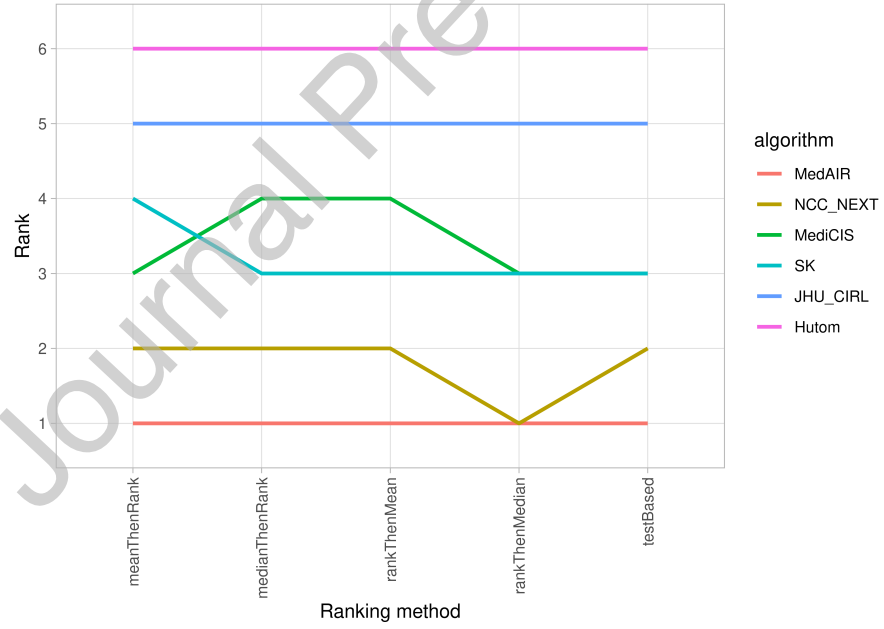


Figure 10: Task 2 recognition ranking stability using the indicated ranking methods.

561 *3.3.3. Task 3: Segmentation-based workflow recognition*

562 Task 3 consisted of recognizing phases, steps, and hand verbs using semantic
563 segmentation data only. First, the results of the segmentation models provided
564 by the participants will be described, and then the workflow recognition models.

565 *Segmentation models:*

566 Table 5 summarizes the methods used by the four participating teams to
perform semantic segmentation.

Team	Hutom *	NCC Next *	SK	MediCIS
Preprocessing	X	X	X	X
Augmentation	X		X	
Model	DeepLabV3+	DeepLabV3+	U-Net	U-Net
Optimizer	Adam	Radam	Adam	Adam
Loss	Equalization v2 & Normsoftmax	cross-entropy	cross-entropy	cross-entropy
Learning Rate	$1e^{-3}$	$1e^{-4}$	$2.4e^{-5}$	$1e^{-4}$

Table 5: Segmentation models used for task 3. Teams that resubmitted models are highlighted with an asterisk. An "X" means that the method performed preprocessing, data augmentation, or is causal.

567

568 Comparison of the IoU values for each class independently and for all classes
569 (Macro) (Table 6) showed that, the IoU varied between 94.0% and 91.1% for
570 Macro. Pegs were the least recognized structure (IoU between 83.9% and 82.3%).
571 Specific sequences with lower performance were not identified.

572 Comparison of the mean IoU values of each team for all classes (Macro) and
573 for each class independently (Table 7) showed similar Macro results for the NCC
574 Next, SK and MediCIS teams (96.9%, 96.4%, and 94.0%, respectively). The
575 Hutom team's Macro IoU was the lowest (85.0%), mainly due to the IoU for
576 pegs (63.3%). Figure 11 presents the ground truth and the segmentation results
577 of each team for one frame.

578 *Workflow models*

	Mean	Median	Max	Min
Background	98.8	98.9	98.9	98.7
Base	96.1	96.2	96.3	95.6
Pegs	83.2	83.1	83.9	82.3
Blocks	91.7	91.7	92.5	90.8
Left tool	94.9	95.3	97.6	87.3
Right tool	94.0	94.5	96.9	88.9
Macro	93.1	93.2	94.0	91.1

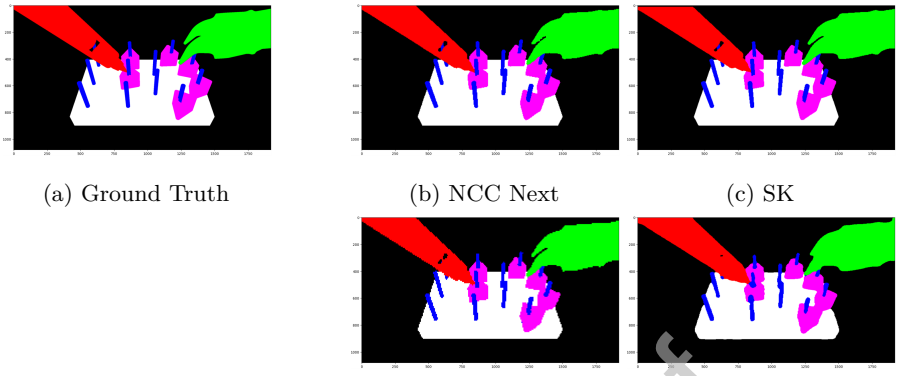
Table 6: Mean Intersection-Over-Union values for all classes of each sequence independently

	Hutom *	NCC Next *	SK	MediCIS
Background	97.7	**99.5**	99.2	98.9
Base	91.4	**98.4**	98.4	96.1
Pegs	63.3	**92.1**	92.0	85.3
Blocks	82.8	96.0	**96.0**	92.2
Left tool	89.3	**98.1**	96.1	96.0
Right tool	85.5	**97.8**	96.7	95.8
Macro	85.0	**96.9**	96.4	94.0

Table 7: Mean Intersection-Over-Union values for all the classes of each team. Teams that resubmitted models are highlighted with an asterisk and best results are in bold.

Table 8 summarizes the methods used by the four participating teams to perform the workflow recognition.

Comparison of the mean AD-Accuracy values for each test sequence (Figure 12) showed that performance decreased from 87.5% to 76.6%. The same two sequences (79 and 54) displayed very low results (67.4% and 65.5%, respectively). Moreover, for all test cases, one model had results lower than 70%.



(a) Ground Truth (b) NCC Next (c) SK

(d) Medicis (e) Hutom

Figure 11: Ground truth (a) and segmentation results for each team (b to e) for one frame.

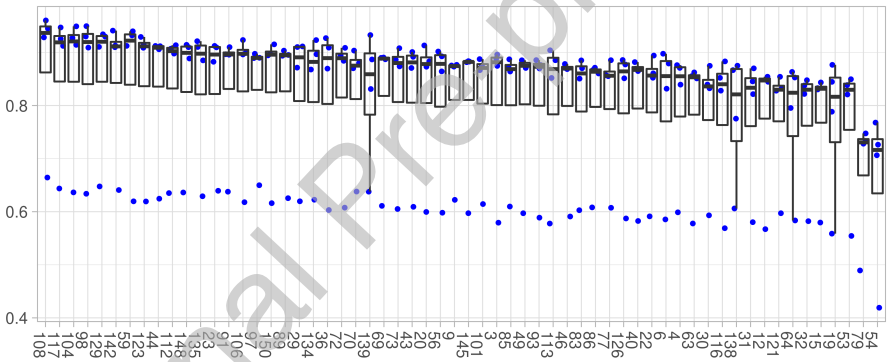


Figure 12: Task 3 recognition AD-Accuracy for each sequence. Each dot represents the AD-Accuracy of one model.

Comparison of the mean AD-Accuracy value for each model indicated that three teams obtained results between 88.5% and 87.2%, whereas the Hutom team had a mean AD-Accuracy value of 60.3% (Figure 13).

Team	Hutom *	NCC Next *	SK	MediCIS
Preprocessing	X	X	X	X
Augmentation	X		X	
Model W	SlowFast50	EfficientNetB7	ResNet18	ResNet50 & MS-TCN++
Optimizer	Adam	Radam	Adam	Adam
Loss	Equalization v2 & Normsoftmax	cross-entropy	cross-entropy	cross-entropy
Learning Rate	$1e^{-3}$	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$
Causal				X

Table 8: Summary of the models used for task 3 (segmentation-based workflow recognition). Teams that resubmitted models are highlighted with an asterisk. An "X" means that the method performed preprocessing, data augmentation, or is causal.

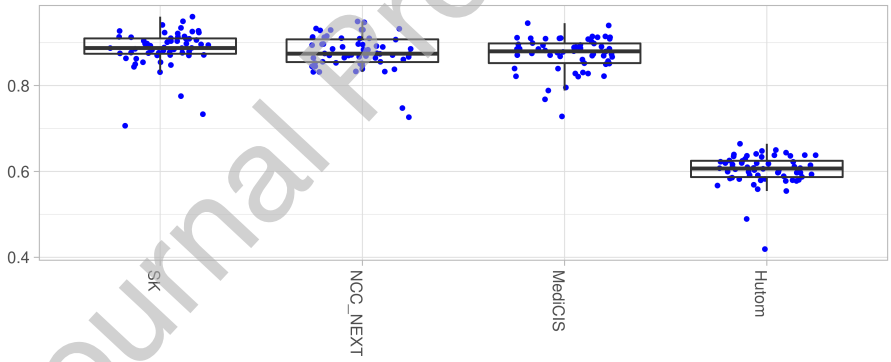


Figure 13: Mean recognition AD-Accuracy for each model for task 3. Each dot represents the AD-Accuracy for one sequence.

The choice of method did not influence the team ranking, except for the second (NCC NEXT) and the third (MediCIS) rank (Figure 14).

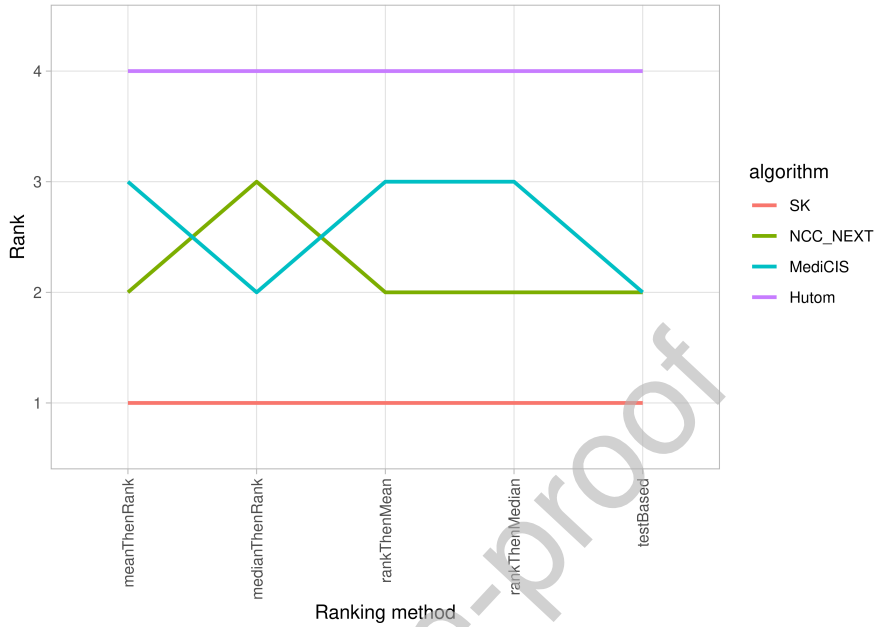


Figure 14: Task 3 recognition ranking stability using the indicated ranking methods.

3.3.4. Task 4: Video/kinematic-based workflow recognition

Task 4 consisted of recognizing phases, steps, and hand verbs using video and kinematic data. Table 9 summarizes the methods used by the six participating teams.

AD-Accuracy values for each sequence were similar to those of the previous tasks (Figure 15). Indeed, performance slightly decreased from 95.1% to 83.1% for most sequences, and was again low for sequences 79 and 54 (81.2% and 76.5%). For this task, the number of outliers was limited.

Team	Hutom *	MedAIR	MMLAB	NCC NEXT *	SK	MediCIS
Preprocessing	X	X	X	X	X	X
Augmentation	X	X			X	
Model	3D ResNet & Bi-LSTM	MRG-Net & CNN	ResNet50 & LSTM	Xception & LightGBM	ResNet18 & Stacked -LSTM	ResNet50 & MS-TCN++
Optimizer	Adam	Adam	Adam	Radam & Gradient Boosting	Adam	Adam
Loss	Equalization v2 & Normsoftmax	cross-entropy	cross-entropy	cross-entropy & MAE	cross-entropy	cross-entropy
Learning Rate	$1e^{-3}$	$1e^{-4}$	$1e^{-3}$ & $1e^{-2}$	$1e^{-1}$ & $5e^{-2}$ & $1e^{-3}$ & $1e^{-4}$	$7.2e^{-4}$ & $1.5e^{-3}$	$1e^{-4}$
Causal			X			

Table 9: Summary of the models used for task 4. Teams that resubmitted models are highlighted with an asterisk. An "X" means that the method performed preprocessing, data augmentation, or is causal.

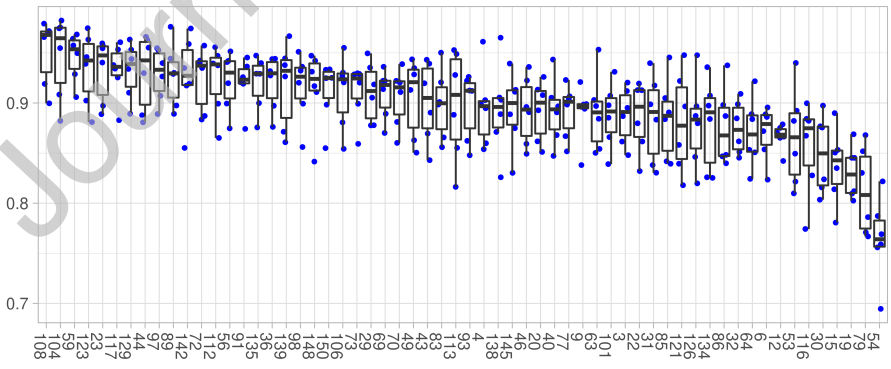


Figure 15: Task 4 recognition AD-Accuracy values for each sequence. Each dot represents the AD-Accuracy for one model.

The NCC NEXT team obtained the best results (Figure 16), with a mean AD-Accuracy of 93.1%, followed by SK, Hutom, and MediCIS teams with results of between 91.6% and 90.2%. For the last two teams, the AD-Accuracy was above 84.5%.

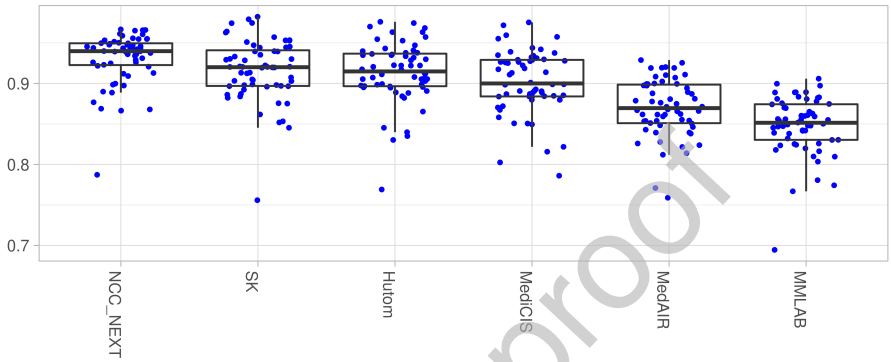


Figure 16: Mean task 4 recognition AD-Accuracy for each team. Each dot represents the AD-Accuracy for one sequence.

The ranking is stable according to the ranking method chosen (Figure 17).

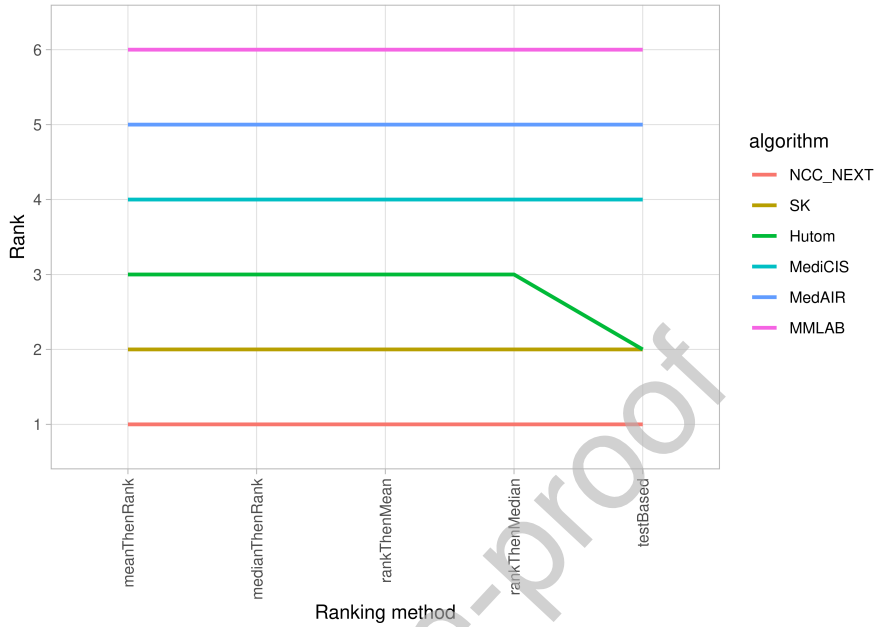


Figure 17: Task 4 recognition ranking stability using the indicated ranking methods.

3.3.5. Task 5: Video/kinematic/segmentation-based workflow recognition

In task 5, teams recognized phases, steps, and hand verbs using video, kinematic and segmentation data. Table 10 summarizes the recognition methods used by the four participating teams. The models to create the segmentation were the same as those described in Table 5.

As for the previous tasks, the mean AD-Accuracy values per sequence (Figure 18) highlighted a slight performance decrease (from 97.2% to 85.9%). Sequences 79 and 54 again displayed the lowest performances (80.8% and 78.0%, respectively).

Team	Hutom *	NCC NEXT *	SK	MediCIS
Preprocessing	X	X	X	X
Augmentation	X		X	
Model	3D ResNet & Bi-LSTM	Xception, EfficientNetB7 & LightGBM	ResNet18 & Staked-LSTM	ResNet50 & MS-TCN++
Optimizer	Adam	Radam & Gradient Boosting	Adam	Adam
Loss	Equalization v2 & Normsoftmax	cross-entropy & MAE	cross-entropy	cross-entropy
Learning Rate	$1e^{-3}$	$1e^{-1}$, $5e^{-2}$, $1e^{-3}$ & $1e^{-4}$	$7.2e^{-4}$, $1.5e^{-3}$ & $1e^{-4}$	$1e^{-4}$
Causal				

Table 10: Models used for task 5. Teams that resubmitted models are highlighted with an asterisk. An "X" means that the method performed preprocessing, data augmentation, or is causal.

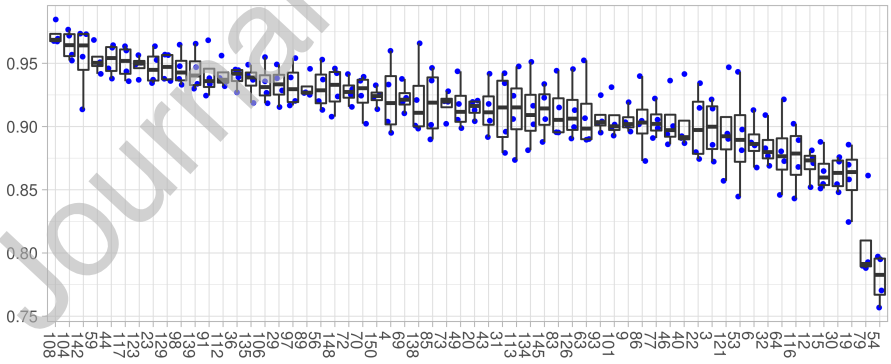


Figure 18: Task 5 AD-Accuracy for each sequence. Each dot represents the AD-Accuracy for one model.

The teams' mean AD-Accuracy values ranged between 93.1% and 89.8% (Figure 19). The SK and Hutom teams displayed very similar results, with 91.4%

and 91.3%, respectively. However, the chosen ranking method did not influence the final rank (Figure 20).

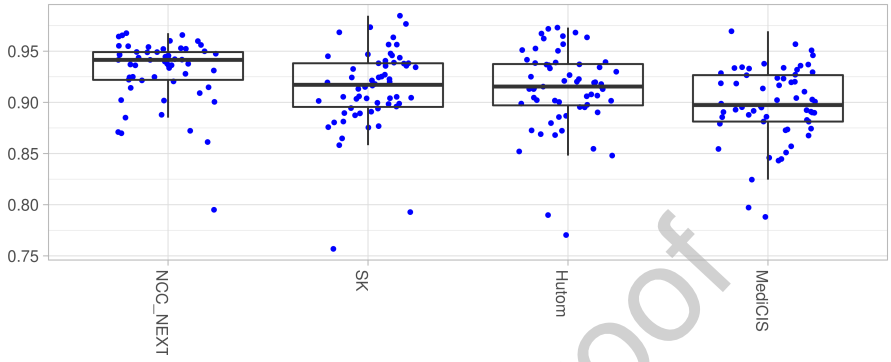


Figure 19: Average task 5 recognition AD-Accuracy for each team. Each dot represents the AD-Accuracy for one sequence.

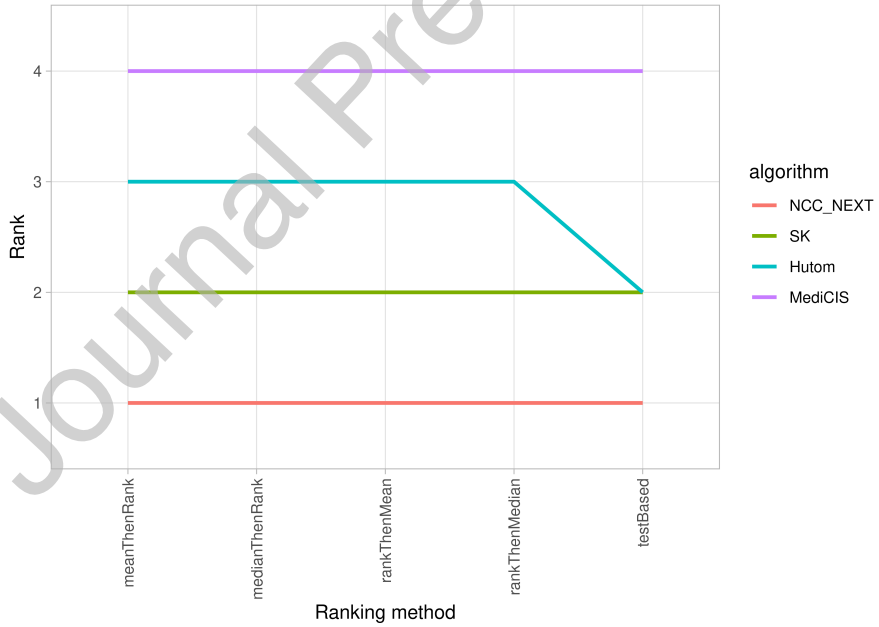


Figure 20: Task 5 ranking stability using the indicated ranking methods.

3.3.6. Workflow recognition results summary

Table 11 summarizes the results of each team for the five tasks. All the best methods displayed mean AD-Accuracy superior to 90%, except for task 3.

Team	Task 1	Task 2	Task 3	Task 4	Task 5
Hutom	90.51 *	84.31	60.28 *	91.33 *	91.27 *
JHU-CIRL		86.45			
MedAIR	84.31 *	**90.72**		86.98	
MMLAB				84.80	
NCC NEXT	87.77 *	90.32	87.71 *	**93.09 ***	**93.09 ***
SK	**90.77**	89.66	**88.51**	91.61	91.37
MediCIS	89.15	89.71	87.22	90.18	89.81

Table 11: Mean AD-Accuracy of each team for the five tasks. The best results are highlighted in bold for each task. Resubmitted models are highlighted with an asterisk.

3.4. Additional analyses

The additional analyses concern four of the seven participating teams: Hutom, NCC Next, SK, and MediCIS. They were the only teams to participate with a combination of the same or similar models used for the unimodal tasks. Although MedAIR team participated in task 4 and the two corresponding unimodal tasks (1 and 2), the models used were too different to allow a model comparison.

3.4.1. Comparison between unimodal and multimodal models

Table 12 presents the results of the statistical analysis. For the four teams, the combination of video and kinematics (task 4) is statistically different than the use of only one modality (tasks 1 and 2). The same statistical differences exist between the combination of the three modalities (task 5) and each modality individually (tasks 1, 2, and 3), with the exception of task 2 and task 5 for the MediCIS team. However, the addition of the segmentation modality (task 5) to the video/kinematic-based (task 4) models was only significant for the MediCIS team.

Team	Hutom	NCC NEXT	SK	MediCIS
T1 <> T4	X	X	X	X
T2 <> T4	X	X	X	X
T1 <> T5	X	X	X	X
T2 <> T5	X	X	X	
T3 <> T5	X	X	X	X
T4 <> T5				X

Table 12: Significant performance differences between unimodal and multimodal tasks. T1 <> T4: comparison of task 1 and task 4; X: significant performance variation (p-value < 0.05).

3.4.2. Execution time

Table 13 presents the execution time for the four teams and each task. For NCC Next team, the duration could not be determined because the predictions were locally written at the end of the Docker image execution. Execution time was highly variable among the teams, with the shortest (except task 2) achieved by the SK team. The shortest execution times overall were obtained for task 2 (3 min for SK and less than 1 minute for the Hutom and MediCIS teams).

Team	Hutom	NCC NEXT	SK	MediCIS
Task 1	56 min	CBD	50 min	202 min
Task 2	< 1 min	CBD	3 min	< 1 min
Task 3	13 550 min	CBD	145 min	725 min
Task 4	57 min	CBD	53 min	203 min
Task 5	13 600 min	CBD	175 min	928 min

Table 13: Execution times to compute the results of the 60 test sequences. CBD: Could not Be Determined

4. Discussion

Accurate surgical workflow recognition is necessary for context-aware computer-assisted surgical systems. The proposed methods obtained good results but were not perfect and the PETRAW data set itself presented several limitations. Specifically, the peg transfer task is significantly easier than a real surgical intervention due to the simpler environment, clearly identifiable objects, static field of view, and constant lighting. In addition, each sequence was performed by the same operator resulting in lower data set variability.

By analyzing the performance of the methods across individual sequences, we observed a gradual decrease in performance, except for two sequences (54 and 79) that displayed very low AD-Accuracy compared to the others regardless of modality. We analyzed these two sequences in detail to understand this poor performance. In sequence 54, the block was dropped twice during the transfer between hands, forcing the operator to catch the block for a second time. In addition, one block got stuck on the peg, forcing the operator to reposition it. Sequence 79 is one of the sequences identified as containing uncertainty (see Section 2.3.3). However, the overlapping steps (by 0.5 seconds) could not entirely explain the low performance, as the overlap was partially compensated by the delay of 0.25 seconds used to compute the AD-Accuracy. In addition, a block got stuck on a peg in this sequence and the order in which the blocks were caught did not correspond to the one used in most sequences. These deviations from the most common workflow might explain the low performance.

For task 1 (video-based recognition), ResNet-based models gave the best results, and the simplest model was ranked first. For task 2 (kinematic-based recognition), LSTM-based methods presented the worst results. For task 3, the two segmentation models used (DeepLabV3 and U-Net), displayed similar IoU values and the differences were probably due to differences in the training characteristics. For workflow recognition, the EfficientNetB7 and ResNet models obtained similar results. For Tasks 4 and 5, the NCC NEXT team's strategy (i.e., using the modality that gave the best results in the unimodal tasks for each

workflow component) provided the best result.

For the segmentation-based recognition task (task 3), the segmentation quality seemed to influence workflow recognition up to a certain threshold. Indeed, the workflow recognition performances of the three teams with Macro IoU values superior to 94.0% were similar (AD-Accuracy between 88.5% and 87.2%), but the ranking was inverted for the two first teams. Conversely, the workflow recognition performance with a Macro IoU value of 85% dropped drastically (60.3%). Additional research is required to fully quantify and understand the degree to which segmentation quality influences workflow recognition since, in this challenge, teams used different combinations of models for the segmentation and workflow recognition components.

For tasks 1 to 4, at least one team submitted a method that could be truly causal. It is important to note that several proposed methods were provably non-causal due to their preprocessing steps and not the core network such as with NCC NEXT (task 3), SK (task 1, 3, 4, 5), and MediCIS (task 2, 4 and 5). Causal methods generally have lower performance than non-causal models. With the exception of task 4, the causal methods displayed performances that were surprisingly close to that of the best method. For example, for task 2, the AD-Accuracy of the best method was 90.7%, compared to 90.3% and 89.7% for the causal methods by NCC NEXT and SK, respectively. Obviously, it is not possible to conclude that causal methods give similar results to acausal models: i) because during the challenge we did not have the two versions of a similar method, ii) due to data simplicity. Nevertheless, the results of the causal methods are promising for developing applications, such as the implementation of automatic reports after training sessions on a virtual simulator.

Among the seven participating teams, four (Hutom, NCC Next, SK, and MediCIS) participated in the multimodal tasks (4 and 5) with a combination of the same or similar models used for the unimodal tasks. In all cases, recognition was improved when several modalities were used (Table 11); however, the addition of segmentation modality decreased the performance. The statistical analysis (Table 12) confirmed a significant performance improvement when using

multimodal models, with the exception of tasks 2 and 5 for the MediCIS team. The performance decrease experienced with the addition of the segmentation modality to the video/kinematic-based models was only significant for the MediCIS team.

Therefore, the combination of video and kinematic (task 4) data gives significantly better results compared with other modality combinations. The results obtained by the MedAIR team could contradict this point because they obtained better results for the kinematic-based recognition task than for the video/kinematic-based one. However, the models they used were very different: a Trans-SVNet and an MRG-Net combined with a CNN respectively. So, in this case, it is difficult to determine if the performance modifications were due to the model or to the modalities used. However, task 4 was more time-consuming than task 2 (53 vs. 3 minutes for SK, 57 vs. less than 1 for Hutom, and 203 vs. less than 1 for MediCIS). One may ask whether it is wise to spend 2,000% to 20,000% more computing time for less than a 3% improvement. The training time should also be taken into account, as it is much more time-consuming [51, 52], but we did not have access to this information. Data storage should also be considered. Video can require a lot of storage space, especially for long surgical interventions. Conversely, kinematic data are less voluminous.

Future work should focus on overcoming the limitations of the current data set by including peg transfer sequences performed by several operators in different systems. Moreover, tests on more realistic data are necessary to validate the finding that kinematic data display the best performances in recognition rate and have less environmental impact thanks to the lowest computation time and storage cost.

Acknowledgements

Authors thanks IRT b<>com for providing the "Surgery Workflow Toolbox [annotate]" software, used for this work.

Statements of ethical approval

All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards. This article does not contain patient data.

Conflict of interest statement

The authors declare that they have no conflict of interest.

References

[1] P. Jannin, M. Raimbault, X. Morandi, and B. Gibaud. Modeling surgical procedures for multimodal image-guided neurosurgery. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 2208, pages 565–572. Springer Verlag, 10 2001.

[2] Florent Lalys and Pierre Jannin. Surgical process modelling: a review. *International Journal of Computer Assisted Radiology and Surgery*, 9(3):495–511, 11 2013.

[3] F Despinoy, D Bouget, G Forestier, C Penet, N Zemiti, P Poignet, and P Jannin. Unsupervised trajectory segmentation for surgical gesture recognition in robotic training. *IEEE Transactions on Biomedical Engineering*, 63(6):1280–1291, 8 2015.

[4] Arnaud Huaulmé, Kanako Harada, Germain Forestier, Mamoru Mitsuishi, and Pierre Jannin. Sequential surgical signatures in micro-suturing task. *International Journal of Computer Assisted Radiology and Surgery*, 13(9):1419–1428, 5 2018.

[5] Germain Forestier, Laurent Riffaud, François Petitjean, Pierre Louis Henaux, and Pierre Jannin. Surgical skills: Can learning curves be computed from

recordings of surgical activities? *International Journal of Computer Assisted Radiology and Surgery*, 13(5):629–636, 5 2018.

[6] S.-Y. Ko, J Kim, W.-J. Lee, and D.-S. Kwon. Surgery task model for intelligent interaction between surgeon and laparoscopic assistant robot. *International Journal of Assitive Robotics and Mechatronics*, 8(1):38–46, 10 2007.

[7] Warren S. Sandberg, Bethany Daily, Marie Egan, James E. Stahl, Julian M. Goldman, Richard A. Wiklund, and David Rattner. Deliberate Perioperative Systems Design Improves Operating Room Throughput:. *Anesthesiology*, 103(2):406–418, 10 2005.

[8] Beenish Bhatia, Tim Oates, Yan Xiao, and Peter Hu. Real-time identification of operating room state from video. In *Proceedings of the National Conference on Artificial Intelligence*, volume 2, pages 1761–1766, 10 2007.

[9] Gwenole Quellec, Mathieu Lamard, Beatrice Cochener, and Guy Cazuguel. Real-Time Task Recognition in Cataract Surgery Videos Using Adaptive Spatiotemporal Polynomials. *IEEE Transactions on Medical Imaging*, 34(4):877–887, 4 2015.

[10] Arnaud Huaulmé, Pierre Jannin, Fabian Reche, Jean-Luc Faucheron, Alexandre Moreau-Gaudry, and Sandrine Voros. Offline identification of surgical deviations in laparoscopic rectopexy. *Artificial Intelligence in Medicine*, 104:1–26, 9 2019.

[11] A Huaulmé, F Despinoy, S A Heredia Perez, K Harada, M Mitsuishi, and P Jannin. Automatic annotation of surgical activities using virtual reality environments. *International Journal of Computer Assisted Radiology and Surgery*, 14(10):1663–1671, 7 2019.

[12] Nicolas Padoy, Tobias Blum, S.-A. Seyed Ahmad S.-A. Seyed Ahmad Ahmadi, Hubertus Feussner, Marie Odile M.-O. Marie Odile M.-O. Berger,

and Nassir Navab. Statistical modeling and recognition of surgical workflow. *Medical Image Analysis*, 16(3):632–641, 1 2010.

[13] Andru P. Twinanda, Sherif Shehata, Didier Mutter, Jacques Marescaux, Michel De Mathelin, and Nicolas Padoy. EndoNet: A Deep Architecture for Recognition Tasks on Laparoscopic Videos. *IEEE Transactions on Medical Imaging*, 36(1):86–97, 2 2016.

[14] L Bouarfa, P P Jonker, and J Dankelman. Discovery of high-level tasks in the operating room. *Journal of Biomedical Informatics*, 44(3):455–462, 10 2011.

[15] A James, D Vieira, B Lo, A Darzi, and G.-Z. Yang. Eye-Gaze Driven Surgical Workflow Segmentation. *Medical Image Computing and Computer-Assisted Intervention MICCAI 2007*, pages 110–117, 11 2007.

[16] Florent Lalys, David Bouget, Laurent Riffaud, and Pierre Jannin. Automatic knowledge-based recognition of low-level tasks in ophthalmological procedures. *International Journal of Computer Assisted Radiology and Surgery*, 8(1):39–49, 4 2012.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-Decem, pages 770–778, 2016.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[20] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[22] Duygu Sarikaya and Pierre Jannin. Surgical Gesture Recognition with Optical Flow only. *arXiv*, 4 2019.

[23] Isabel Funke, Sebastian Bodenstedt, Florian Oehme, Felix von Bechtolsheim, Jürgen Weitz, and Stefanie Speidel. Using 3D Convolutional Neural Networks to Learn Spatiotemporal Features for Automatic Surgical Gesture Recognition in Video. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11768 LNCS, pages 467–475, 2019.

[24] Robert DiPietro and Gregory D. Hager. Automated Surgical Activity Recognition with One Labeled Sequence, 10 2019.

[25] Arnaud Huaulmé, Duygu Sarikaya, Kévin Le Mut, Fabien Despinoy, Yonghao Long, Qi Dou, Chin-Boon Chng, Wenjun Lin, Satoshi Kondo, Laura Bravo-Sánchez, Pablo Arbeláez, Wolfgang Reiter, Manoru Mitsuishi, Kanako Harada, and Pierre Jannin. MIcro-surgical anastomose workflow recognition challenge report. *Computer Methods and Programs in Biomedicine*, 212:106452, 11 2021.

[26] Yong-Hao Long, Jie-Ying Wu, Bo Lu, Yue-Ming Jin, Mathias Unberath, Yun-Hui Liu, Pheng-Ann Heng, and Qi Dou. Relational Graph Learning on Visual and Kinematics Embeddings for Accurate Gesture Recognition in Robotic Surgery. *arXiv*, 2020.

[27] Yidan Qin, Max Allan, Yisong Yue, Joel W. Burdick, and Mahdi Azizian. Learning Invariant Representation of Tasks for Robust Surgical State Estimation. *arXiv*, 2 2021.

[28] S.A Heredia Perez, Kanako Harada, and Mamoru Mitsuishi. Haptic As-

sistance for Robotic Surgical Simulation. *27th Annual Congress of Japan Society of Computer Aided Surgery*, 20(4):232–233, 11 2018.

[29] O Dergachyova, D Bouget, A Huaulmé, X Morandi, and P Jannin. Automatic data-driven real-time segmentation and recognition of surgical workflow. *International Journal of Computer Assisted Radiology and Surgery*, 10 2016.

[30] Lena Maier-Hein, Matthias Eisenmann, Annika Reinke, Sinan Onogur, Marko Stankovic, Patrick Scholz, Tal Arbel, Hrvoje Bogunovic, Andrew P Bradley, Aaron Carass, Carolin Feldmann, Alejandro F Frangi, Peter M Full, Bram van Ginneken, Allan Hanbury, Katrin Honauer, Michal Kozubek, Bennett A Landman, Keno März, Oskar Maier, Klaus Maier-Hein, Bjoern H Menze, Henning Müller, Peter F Neher, Wiro Niessen, Nasir Rajpoot, Gregory C Sharp, Korsuk Sirinukunwattana, Stefanie Speidel, Christian Stock, Danail Stoyanov, Abdel Aziz Taha, Fons van der Sommen, Ching-Wei Wang, Marc-André Weber, Guoyan Zheng, Pierre Jannin, and Annette Kopp-Schneider. Why rankings of biomedical image analysis competitions should be interpreted with care. *Nature Communications*, 9(1):5217, 12 2018.

[31] Manuel Wiesenfarth, Annika Reinke, Bennett A. Landman, Matthias Eisenmann, Laura Aguilera Saiz, M. Jorge Cardoso, Lena Maier-Hein, and Annette Kopp-Schneider. Methods and open-source toolkit for analyzing and visualizing challenge results. *Scientific Reports*, 11(1):2369, 12 2021.

[32] Pierre Jannin. Towards responsible research in digital technology for health care. 9 2021.

[33] Liang Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11211 LNCS, pages 833–851, 2018.

[34] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[35] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2019-Octob, pages 6201–6210, 2019.

[36] Mike Schuster and Kuldip K Paliwal. Bidirectional Recurrent Neural Networks. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, 45(11), 1997.

[37] Xinlei Chen and Kaiming He. Exploring Simple Siamese Representation Learning. 2020.

[38] Jingru Tan, Xin Lu, Gang Zhang, Changqing Yin, and Quanquan Li. Equalization Loss v2: A New Gradient Balance Approach for Long-tailed Object Detection. 12 2020.

[39] Andrew Zhai and Hao Yu Wu. Classification is a Strong Baseline for Deep Metric Learning. *30th British Machine Vision Conference 2019, BMVC 2019*, 11 2018.

[40] Robert Dipietro, Colin Lea, Anand Malpani, Narges Ahmidi, S. Swaroop Vedula, Gyusung I. Lee, Mija R Lee, and Gregory D Hager. Recognizing surgical activities with recurrent neural networks. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9900 LNCS, pages 551–558, 2016.

[41] Xiaojie Gao, Yueming Jin, Yonghao Long, Qi Dou, and Pheng Ann Heng. Trans-SVNet: Accurate Phase Recognition from Surgical Videos via Hybrid Embedding Aggregation Transformer. *Lecture Notes in Computer Science*

(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12904 LNCS:593–603, 3 2021.

[42] François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:1800–1807, 10 2016.

[43] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the Variance of the Adaptive Learning Rate and Beyond. 8 2019.

[44] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems*, 30, 2017.

[45] Mark Everingham, Luc Van Gool, Christopher K.I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision 2009 88:2*, 88(2):303–338, 9 2009.

[46] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:10691–10700, 5 2019.

[47] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2623–2631, 7 2019.

[48] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9351:234–241, 2015.

[49] Shijie Li, Yazan Abu Farha, Yun Liu, Ming-Ming Cheng, and Juergen Gall. MS-TCN++: Multi-Stage Temporal Convolutional Network for Action

Segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:3570–3579, 6 2020.

[50] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 9 2014.

[51] David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon Emissions and Large Neural Network Training.

[52] Emma Strubell, Ananya Ganesh, and Andrew Mccallum. Energy and Policy Considerations for Deep Learning in NLP. 2019.