SVIP: Sequence VerIfication for Procedures in Videos – Supplementary –

Anonymous CVPR submission

Paper ID 5042

This document is the supplemental material of our CVPR2022 paper — *SVIP: Sequence VerIfication for Procedures in Videos*, and is arranged as follows:

1). The first section contains some complementary information of our proposed CSV dataset, *e.g.*, data gathering, annotations, and statistics information.

2). The second section gives more examples of scoring and a demo of another application, early warning.

A. CSV Dataset

The existing action datasets can hardly support our task due to the following reasons: i) some datasets focus on single actions and don't provide procedure videos; ii) some other datasets which contain procedure videos target other tasks such as action segmentation and action localization, *i.e.*, they focus on the understanding of a single video rather than the verification of two videos, which leads to the lack of videos for similar procedures. However, the verification task indubitably requires a great number of videos that perform similar but slightly different step sequences for training. For the above reasons, we collect a new action verification dataset to support our proposed task. In this section, we firstly describe the gathering process of the dataset, then give the annotation details of the videos, and finally demonstrate the statistical information of the dataset.

A.1. Data Gathering

The dataset is recorded with the participation of 82 volunteers, whose ages range from 21 to 28, for performing scripted action sequences. Considering the constraints of venues, props, and personnel, we record videos of participants first setting up the equipment to perform a chemical experiment and then conducting that experiment. The specific process of recording is divided into the following steps: i) we firstly predefine 14 chemical experiment tasks, each of which contains consists of 5 procedures with a few step-level divergences, which will be detailed stated in Sec. A.2; ii) the volunteers are required to remember these predefined operations and equip with a head-mounted camera (shown in Figure 1); iii) after the camera start working, the volunteers are asked to perform the predefined action sequences and put hands on the table or their sides when finished, and then the recording will be stopped. In this way, the integrity of procedures in the videos gets guaranteed.

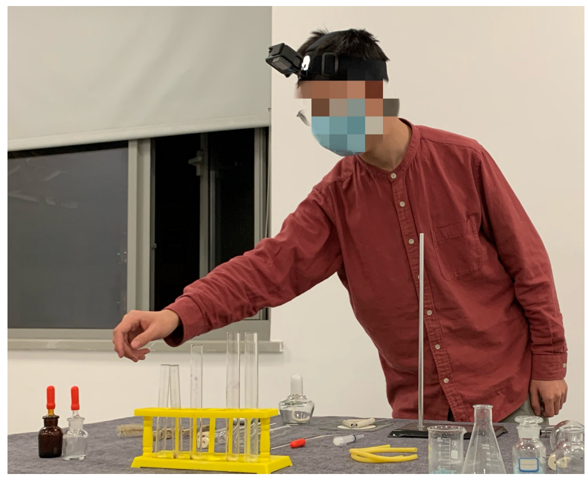

Figure 1. Head-mounted device used in data recording.

Following the collected method of [1], we choose GoPro HERO4 Black with an adjustable mounting such that the camera device can adjust to an appropriate pose with the variance of wearers' height, which provides multi-angle views and makes that each video contains interactions between the volunteers' hands and apparatus on the same experiment table. Besides, to ensure the stability and quality of the video, the camera is connected to a monitoring tablet via Bluetooth in order to monitor the quality of the recorded video at any time. Once a mistake occurs, the video will be discarded and re-shot. When shooting, the camera is set to the linear field of view, $24fps$, and the resolution of 1920×1080. Stereo audio is captured but discarded since almost all procedures proceed silently, and the sounds in videos will cause irrelevant noises.

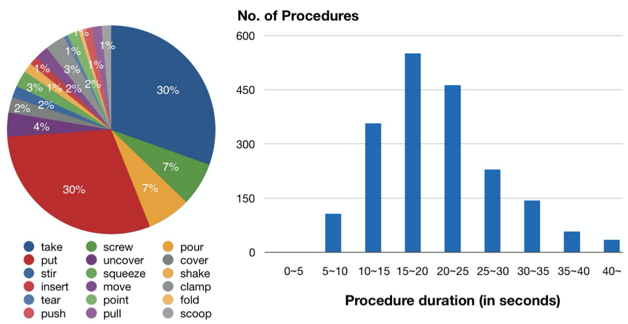

Figure 2. **Left**: Percentage of occurrences of each atomic-level action; **Right**: The histogram of step durations.

A.2. Action Sequence Annotations

In order to cater to our objective of verifying similar procedures with few step-level differences, we design fourteen different tasks in chemical experiments, and each enumerates all step-level transformations, *e.g.*, additions, deletions, order exchange of steps. A step is defined as an action-object interaction whose label is always a combination of a verb, a noun, and sometimes prepositions. We label all procedures as $1.1 \sim 1.5,\ 2.1 \sim 2.5,\ \cdots,\ 14.1 \sim 14.5$, totally 14 tasks, 70 labels. Note that we annotate each video only with a single serial number indicating the category of the procedure in the video, but without any temporal annotations, including the start and end frame of steps. Take the first task of procedures which is about screwing the test tube onto the iron stand and pouring water into the test tube as an example.

- *1.1*: **take** (up the iron clamp) - **screw** (the iron clamp) - **take** (up the test tube) - **screw** (the iron clamp) - **take** (up the conical flask) - **pour** (the conical flask) - **put** (down the conical flask)
- *1.2*: **take** (up the iron clamp) - **take** (the a test tube) - **screw** (the iron clamp) - **screw** (the iron clamp) - **take** (up the conical flask) - **pour** (the conical flask) - **put** (down the conical flask)
- *1.3*: **take** (up the test tube) - **take** (up the conical flask) - **pour** (the conical flask) - **put** (down the conical flask) - **take** (up the iron clamp) - **screw** (the iron clamp) - **screw** (the iron clamp)
- *1.4*: **take** (up the iron clamp) - **screw** (the iron clamp) - **take** (up the conical flask) - **put** (down the conical flask) - **take** (up the test tube) - **screw** (the iron clamp)
- *1.5*: **take** (up the iron clamp) - **screw** (the iron clamp) - **take** (up the test tube) - **screw** (the iron clamp) - **take** (up the conical flask) - **put** (down the conical flask) - **take** (up the conical flask) - **pour** (the conical flask) - **put** (the conical flask)

As illustrated above, compared to the *1.1*, *1.2* and *1.3* disturb the order of actions; *1.4* not only changes the order, but also deletes the **pour** action; and for *1.5*, it inserts **take** - **put** actions into the standard one.

The first group of procedures, which is a microcosm of the whole dataset, shows that most procedures differ in step order. The reason that we are so concerned about the order is that most action sequences will be unmeaning, sometimes even dangerous, if the order changes. For example, it is meaningless or even ridiculous to apply soap to hands after finishing washing hands.

A.3. Statistical Information

Figure 2 shows some statistics of our dataset. As illustrated, we have 18 atomic-level actions with different frequencies in total, among which **take** and **put** are the two most common actions. This makes sense since taking up or putting down something is also extremely common in reality. By interacting one action with different objects, we have 106 steps in total (listed in Figure 3). The videos' length varies from 5.63s to 58.43s due to the diversity in complexity among procedures and individual differences of participants, such as movement habits, the memory of the action sequence as well as familiarity with the operations. Totally, we collect around 960,000 images of over 1,940 videos across 70 different kinds of procedures. On average, each video lasts 20.58 seconds, contains 495.85 frames, and consists of 9.53 steps.

B. Demos

B.1. Scoring

In this section, we demonstrate more examples as the scoring demo, which is detailed in Section 5.6 of the main body of this paper. For each dataset, we show two positive and two negative pairs, a total of eight videos with their procedure label. We can find Figure 6 has different procedure annotation from Figure 7 and 8, since the original COIN dataset [3] has temporal annotation for each step but Diving48 [2] and CSV doesn't. It is worth noticing that V_3 and V_4 in Figure 7 perform the same diving sequence but recorded from different directions of the athlete but still outputs a high matching score.

B.2. Early Warning

In addition to scoring, the sequence verification task can also be applied in early warning. The system is required to alarm whenever it detects the occurrence of an unexpected step. Thus, how to detect atomic-level actions in real-time and how to compare the incomplete input procedure with the complete reference procedure would be the main difficulties of this promising task, which is also our future research direction.

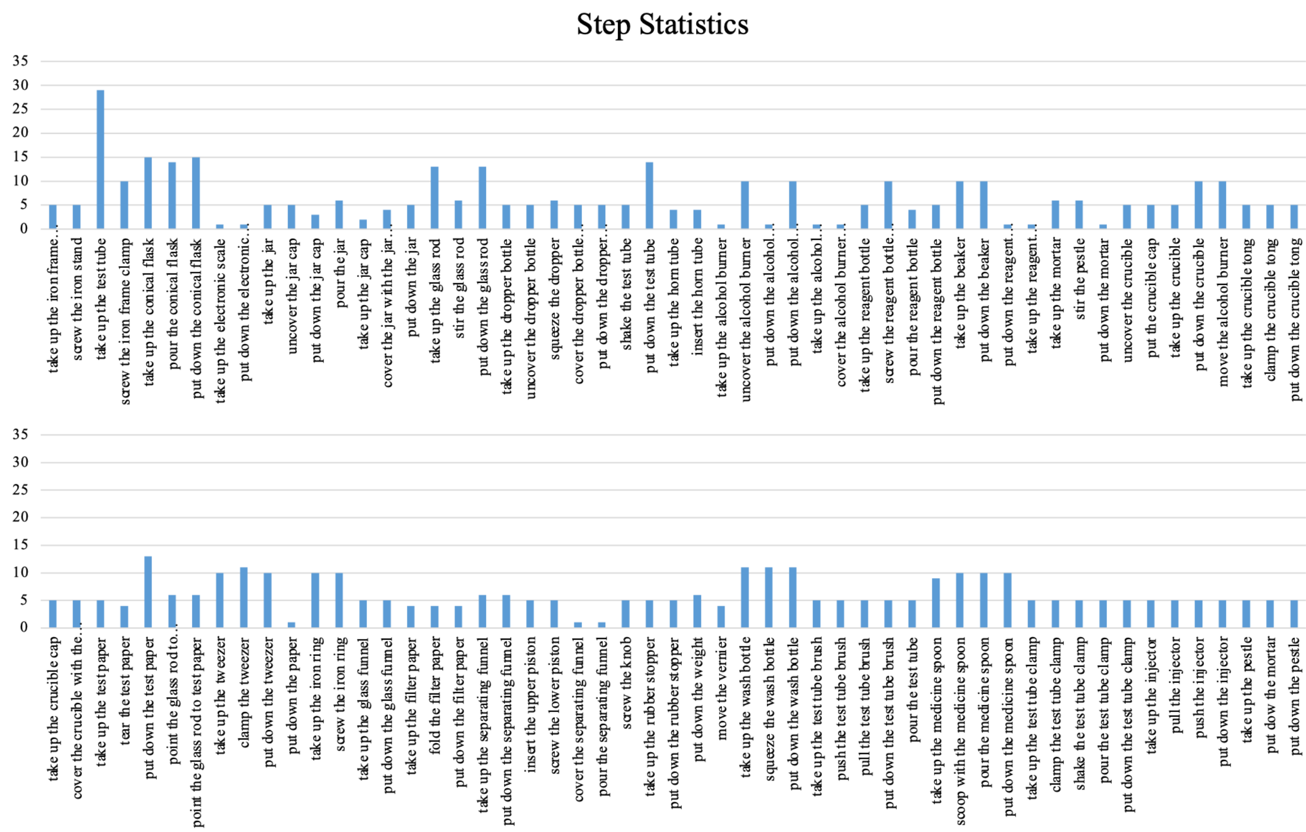

Figure 3. Step statistics. We list all 106 steps with their frequency of occurrence.

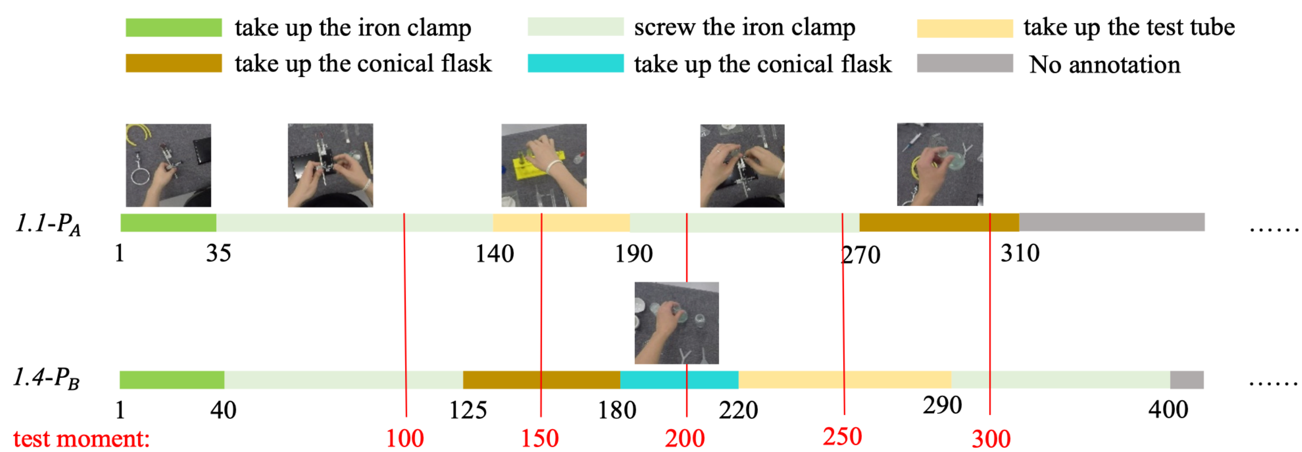

Figure 4. The temporal annotations of the exampled procedures. Blocks in the same color means that the corresponding clips of frames are annotated by the same step. The numbers on both sides of the block are the index of start and end frames of this action.

However, the main body of this paper is to solve the verification problem of two complete procedures, which we named off-line verification. Here, we simply extend it to on-line sequence verification, where we can verify whether the input procedure is consistent with the reference in an on-line video stream. We design the following baseline. We take videos with labels *1.1* and *1.4*, which are performed by two participants P_A, P_B, for demonstration. According to the detailed illustration in Section A.2, sequence *1.4* and sequence *1.1* are the same in the first three steps but are different in the fourth step. Note that although the third step are the same, the objects they interact with are different.

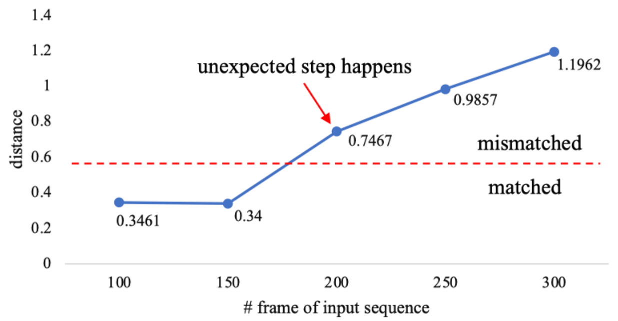

Figure 5. The evaluation result of the exampled video pair with on-line verification baseline.

However, such differences may be difficult for the model to recognize due to the light transmittance of glass products. The following is the specific description of the on-line action verification baseline.

Given a t-frame test procedure P_{test} and the corresponding reference procedure P_0, and assume that it takes similar time intervals for each individual to perform the same step (this assumption is the basis of the baseline). Then we can assume that $P_0[1 : t \pm k]$ (the first $t \pm k$ frames of the reference procedure) is expected to perform the same step-sequence as $P_{\text{test}}[1 : t]$ does if they are labeled the same, where k is the time window size ($k = 30$ in our experiment). For each $P_0[1 : t+i]$, $-k \leq i \leq k$, we calculate the l_2 distance between $P_0[1 : t+i]$ and $P_{\text{test}}[1 : t]$ in the feature space f and average them over $2k+1$ cases as followed:

$$\frac{\sum_{i=-k}^{k} \|f(P_{\text{test}}[1:t]) - f(P_0[1:t+i])\|_2^2}{2k+1}$$

Specifically, we stipulate all the frames of procedure *1.1* performed by P_A as the complete reference procedure, and the first 100/150/200/250/300 frames of procedure *1.4* performed by P_B as incomplete test procedures, the temporal annotation of these frames are given in Figure 4.

Figure 5 shows our experimental results. The blue line represents for the calculated l_2 distance in the feature space f between *1.4*-P_B and *1.1*-P_A with different number of input frames. For the convenience of explanation, we notate the number of input test frames as i. When $i = 100$, the value of *distance* remains relatively low. This is because both the first 100 frames of *1.4*-P_B and the similar amount of frames of *1.1*-P_A perform the same steps. When $i = 150$, note that although the objects interacted by the third step **take** around frame 150 are different in *1.4*-P_B and *1.1*-P_A (conical flask and test tube), such glass products are hard to distinguish by the model, which also leads to the small value of *distance*. When $i = 200$, the step in *1.4*-P_B is significantly different from the step in *1.1*-P_A. Thus, the value of *distance* rises rapidly. Besides, the broken line goes higher when $i = 250$ or 300 since more unmatched steps are included. We can easily catch the unexpected step in an on-line video stream through the huge jump of the line.

According to above, when we choose an appropriate threshold of *distance*, the *1.4*-P_B *vs.* *1.1*-P_A pair is verified until the number of input frames achieves 200, the moment when the unmatched step occurs, which satisfies the requirement of on-line action verification. This section states a coarse mechanism for on-line action verification and evaluates a toy sample based on that, which can be applied in the field of early warning. We hope that this brick cast away can attract a jode, *i.e.*, makes more researchers study this challenging but promising task.

References

[1] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736, 2018. 1

[2] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 513–528, 2018. 2

[3] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1207–1216, 2019. 2

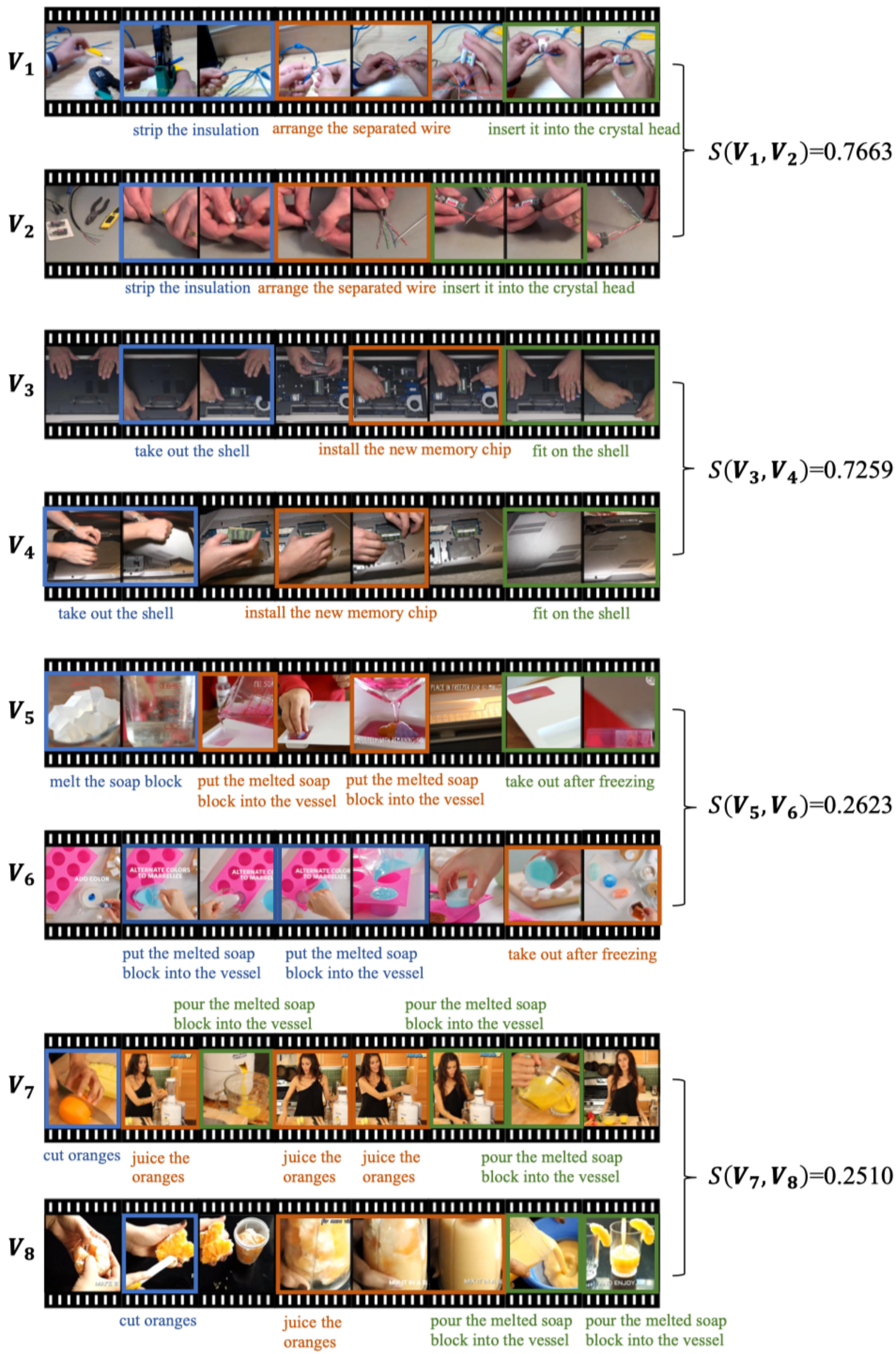

Figure 6. COIN-SV scoring example.

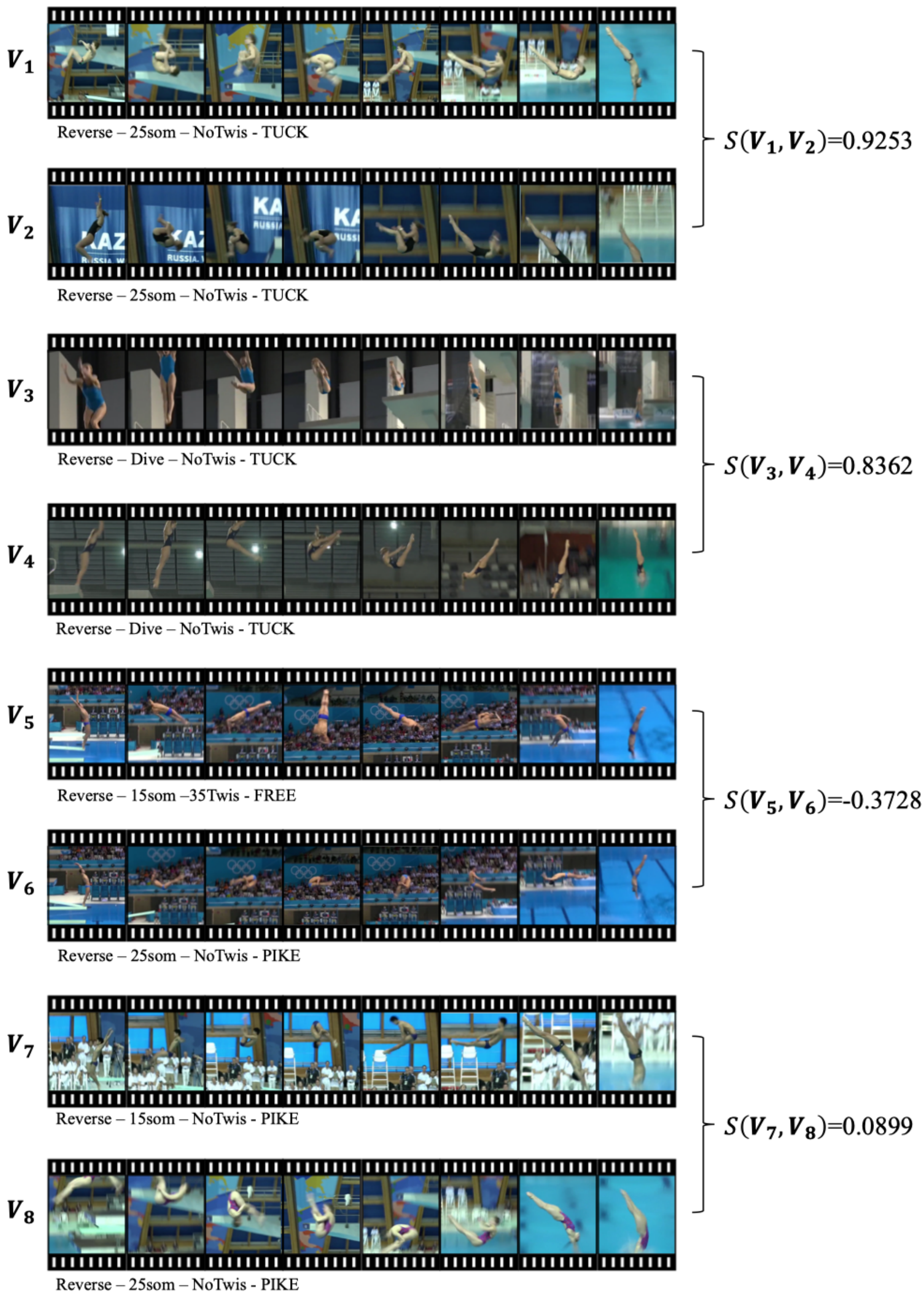

Figure 7. Diving48-SV scoring example.

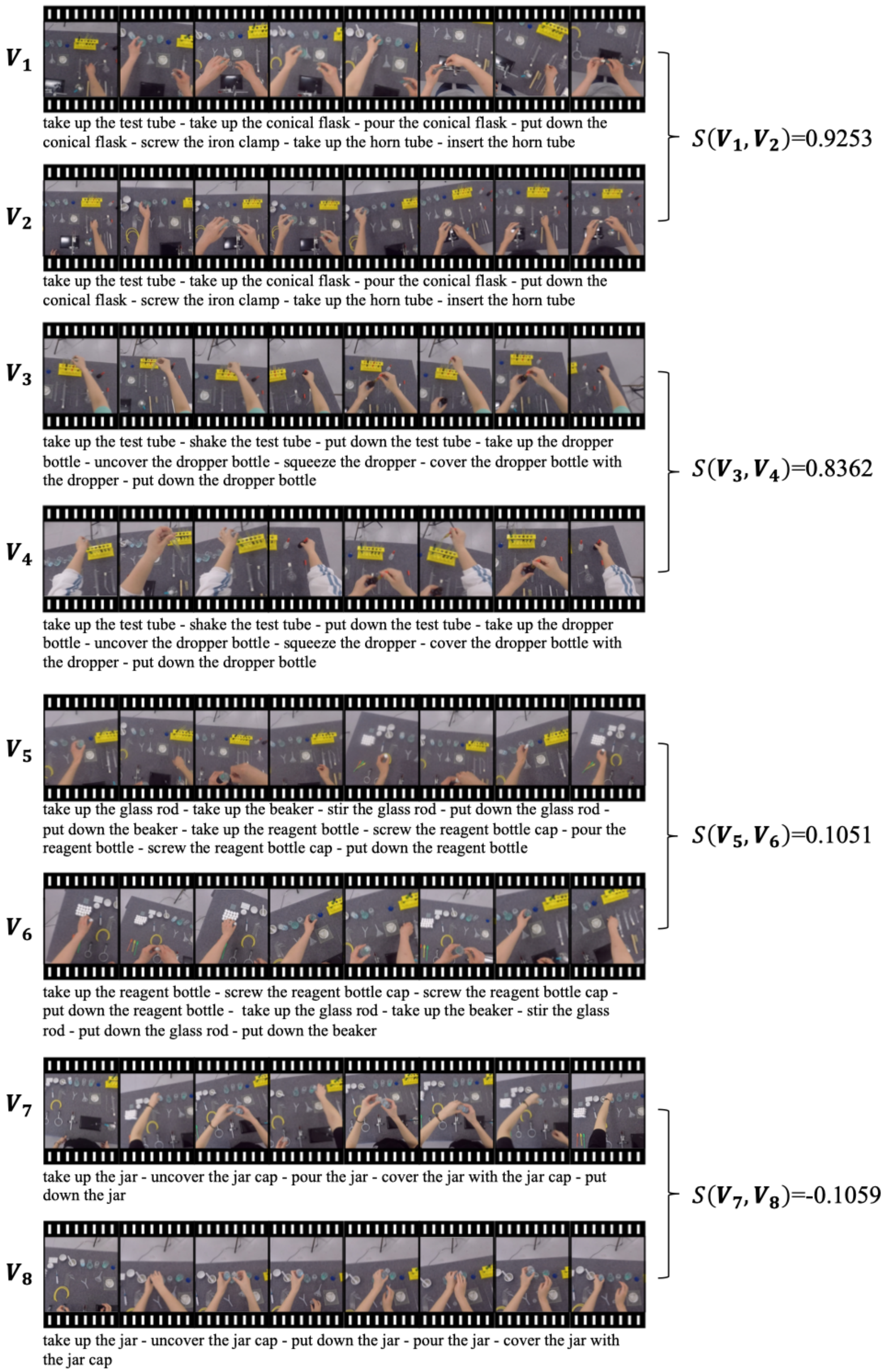

Figure 8. CSV scoring example.