III. Automated Evaluation

In this section, we described the procedure we used to evaluate `CAPdroid` in terms of its performance automatically. Since our approach consists of two main automated steps to obtain the actions from the recordings, we evaluate these phases accordingly, including Action Segmentation (Section II-B), and Action Attribute Inference (Section II-C). Consequently, we formulated the following two research questions:

- **RQ1**: How accurate is our approach in segmenting action clips from GUI recordings?
- **RQ2**: How accurate is our approach in inferring action attributes from clips?

To perform the evaluation automatically, we leveraged the existing automated app exploration tool Droidbot [43] to collect GUI recordings with ground-truth actions. In detail, we first collected 439 top-rated Android apps from Google Play covering 14 app categories (e.g., news, tools, finance, etc.). Each app was run for 10 minutes by Droidbot to automatically explore app functionalities by simulating user actions on the GUI. The simulated actions, including operation time, types, locations, etc, were dumped as metadata, representing the ground truth. Meanwhile, we captured a screen recording to record the actions for each app at 30 fps. As discussed in Section II-A, users may use different indicators to depict their touches. To make our recordings as similar to real-world recordings as possible, we adopted different touch indicators to record actions, including 181 default, 152 cursor, and 106 custom. In total, we obtained 439 10-min screen recordings as the experimental dataset for the evaluation.

A. RQ1: Accuracy of Action Segmentation

Experimental Setup. To answer RQ1, we evaluated the ability of our `CAPdroid` to precisely segment the recordings into action clips and accurately classify the actions. To accomplish this, we utilized the metadata of action operation time as the ground-truth. During preliminary observation with many recordings, we found that, due to the delay between commands and operations on the device, it may have small time-frame differences between the ground-truth and the recorded actions. To avoid these small differences, we broadened the ground-truth of the actions by 5 frames. In total, we obtained 12k *TAP*, 4k *SCROLL*, and 1k *INPUT* clips from 439 screen recordings.

Metrics. We employed two widely-used evaluation metrics, e.g., video segmentation F1-score, and accuracy. To evaluate the precision of segmenting the action clips from recordings, we adopted video segmentation F1-score [44], which is a standard video segmentation metric to measure the difference between two sequences of clips that properly accounts for the relative amount of overlap between corresponding clips. Consider the clips segmented by our method (c_{our}) and ground truth (c_{gt}), vs-score is computed as $\frac{2|c_{our} \cap c_{gt}|}{|c_{our}|+|c_{gt}|}$, where $|c|$ denotes the duration of the clip. The higher the score value, the more precise the method can segment the video. We further adopted accuracy to evaluate the performance of our approach to discriminate action types from clips. The higher the accuracy score, the better the approach can classify the actions.

Method	TAP		SCROLL		INPUT		Overall	
	VS	Acc	VS	Acc	VS	Acc	VS	Acc
ABS	0.56	0.69	0.59	0.69	0.67	0.73	0.61	0.71
HIST	0.71	0.80	0.62	0.71	0.75	0.84	0.70	0.79
SIFT	0.61	0.71	0.60	0.73	0.63	0.79	0.62	0.75
SURF	0.55	0.71	0.59	0.72	0.60	0.77	0.58	0.74
EDGE	0.61	0.75	0.55	0.70	0.66	0.78	0.61	0.75
Ours	**0.81**	**0.89**	**0.83**	**0.92**	**0.90**	**0.97**	**0.84**	**0.93**

TABLE II: Performance comparison of action segmentation. "VS" denotes the video segmentation F1-score, and "Acc" denotes the accuracy of action classification.

Baselines. To demonstrate the advantage of using SSIM as the image similarity metric to segment actions from GUI recordings, we compared it with 5 image-processing baselines, including pixel level (e.g, absolute differences ABS [45], color histogram HIST [46]), structural level (e.g., SIFT [47], SURF [48]), and motion-estimation level (e.g., edge detection EDGE [49]). Due to the page limit, we omitted the details of these well-known methods.

Results. Table II shows the overall performance of all baselines. The performance of our method is much better than that of other baselines, i.e., 20%, 17% boost in video segmentation F1-score and accuracy compared with the best baseline (HIST). Although HIST achieves the best performance in the baselines, it does not perform well as it is sensitive to the pixel value. This is because the recordings can often have image noise due to fluctuations of color or luminance. The image similarity metrics based on structural level (i.e., SIFT, SURF) are not sensitive to image pixel, however, they are not robust to compare GUIs. This is because, unlike images of natural scenes, features in the GUIs may not distinct. For example, a GUI contains multiple identical checkboxes, and the duplicate features of checkboxes can significantly affect similarity computation. Besides, motion-estimation baseline (EDGE) cannot work well in segmenting actions from GUI recordings, as GUI recordings are artificial artifacts with different rendering processes. In contrast, our method using SSIM achieves better performance as it takes similarity measurements in many aspects from spatial and pixel, which allows for a more robust comparison.

Our method also makes mistakes in action segmentation due to two reasons. First, we wrongly segment one action clip into multiple ones due to the unexpected slow resource loading, e.g., one clip for the GUI transition of a user action, and the other clip for the GUI's resource loading. Second, some GUIs may contain animated app elements such as advertisements or movie playing, which will change dynamically, resulting in mistake action segmentation and classification.

B. RQ2: Accuracy of Action Attribute Inference

Experimental Setup. To answer RQ2, we evaluated the ability of our `CAPdroid` to accurately infer the action attributes from the segmented clips. To accomplish this, we