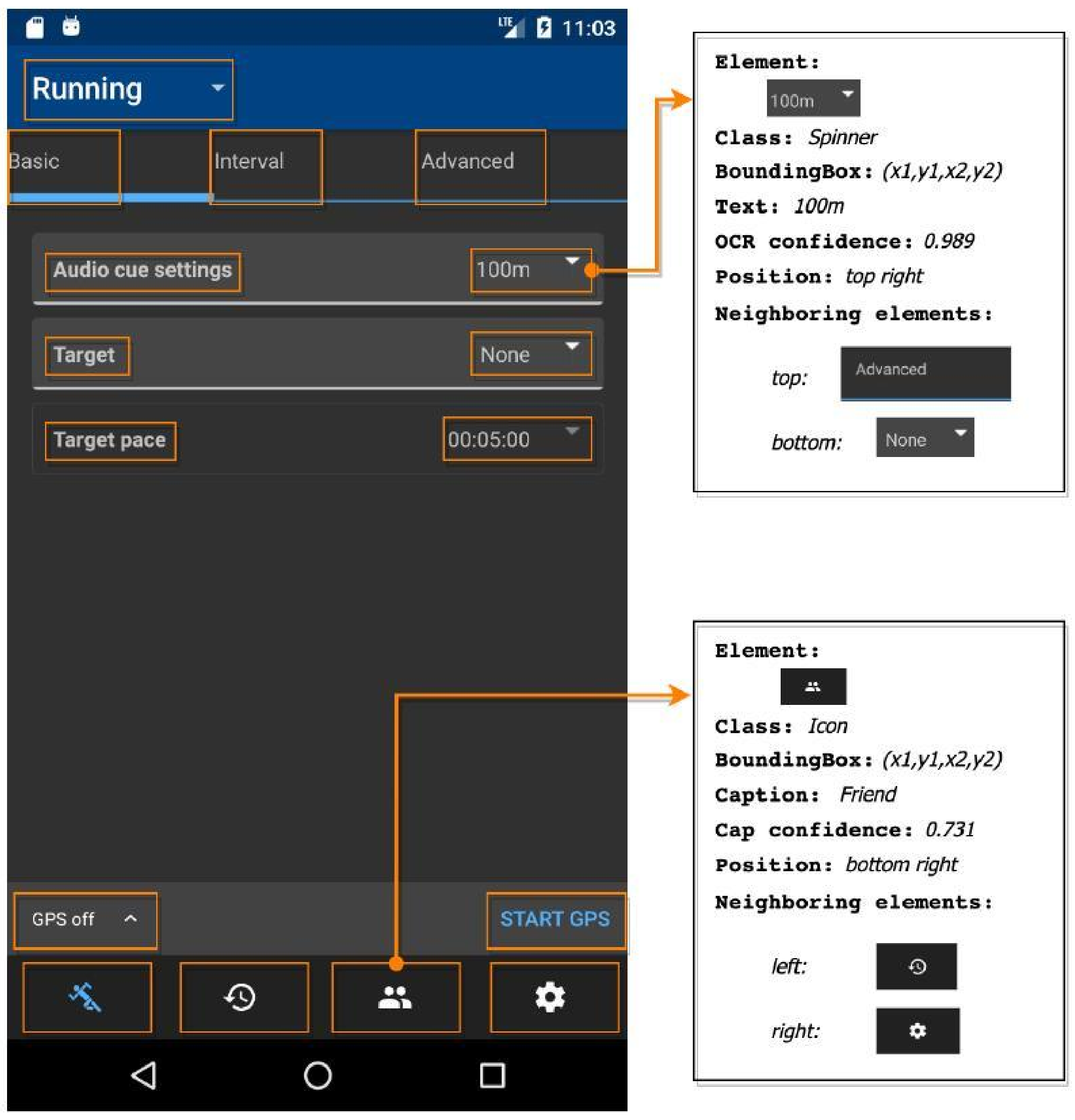
| Action | Id | Condition | Template | Example |
|---|---|---|---|---|
| *TAP* | 1 | $(obj_{text/caption} \neq NULL) \wedge (obj_{confid} > \alpha)$ | Tap [$obj_{text}$] [$obj_{class}$] | Tap “OK” button |
| | 2 | $(obj_{text/caption} \neq NULL) \wedge (\beta < obj_{confid} < \alpha)$ | Tap [$obj_{text}$] [$obj_{class}$] at [$obj_{position}$] | Tap “menu” icon at top left corner |
| | 3 | $(obj_{text/caption} == NULL) \vee (obj_{confid} < \beta)$ | Tap the [$obj_{class}$] [$nbr_{relation}$] [$nbr_{text}$] | Tap the checkbox next to “Dark Mode” |
| *SCROLL* | 4 | $obj_{text} \neq NULL$ | Scroll [*direction*] [*offset*] of the screen to [$obj_{text}$] | Scroll down half of the screen to “Advanced Setting” |
| | 5 | $obj_{text} == NULL$ | Scroll [*direction*] [*offset*] of the screen | Scroll up a quarter of the screen |
| *INPUT* | 6 | $(obj_{text/caption} \neq NULL) \wedge (obj_{confid} > \alpha)$ | Input [*text*] in the [$obj_{text}$] edittext | Input “100” in the “Amount” edittext |
| | 7 | $(obj_{text} == NULL) \vee (obj_{confid} < \alpha)$ | Input [*text*] in the edittext [$nbr_{relation}$] [$nbr_{text}$] | Input “John” in the edittext below “Name” |

TABLE I: Description template, where “*obj*” and “*nbr*” denote the GUI element and its neighbor, $\alpha$ and $\beta$ denote high- and low-confidence element.



Element:
100m
Class: *Spinner*
BoundingBox: *(x1,y1,x2,y2)*
Text: *100m*
OCR confidence: *0.989*
Position: *top right*
Neighboring elements:
*top:* Advanced
*bottom:* None

Element:
Class: *Icon*
BoundingBox: *(x1,y1,x2,y2)*
Caption: *Friend*
Cap confidence: *0.731*
Position: *bottom right*
Neighboring elements:
*left:*
*right:*

Fig. 7: Example of GUI understanding.

and the “None” element at the *bottom*. Note that the “Audio cue settings” element is omitted due to large spacing, which is consistent with human viewing.

*2) Subtitle Creation:* The main instruction of interest is to create a clear and concise subtitle description based on $\{action, object\}$. The global GUI information is further used to complement the description by $\{position, relationship\}$. Based on the $action$ obtained in Section II-B, the attribute of $object$ inferred in Section II-C, and the corresponding GUI element information retrieved in Section II-D1, we propose description templates for *TAP*, *SCROLL*, *INPUT*, respectively. A summary of description templates can be seen in Table I.

For *TAP* action, the goal of the description should be clear and concise, e.g., tap “OK” button. However, we find that this simple description may not articulate all *TAP* actions due to two reasons. First, the text and caption of *object* are prone to errors or undetected, as the OCR-obtained text and the caption-obtained annotation are not 100% accurate. Second, there may be multiple *objects* with the same text on the GUI. To resolve this, we set up an *object* confidence value $obj_{confid}$ as:

$$obj_{confid} = \begin{cases} OCR_{confid} & \text{if } obj_{text} \text{ is unique in GUI} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $OCR_{confid}$ denotes the confidence predicted by OCR. Note that the confidence value of icon *object* is calculated likewise by captioning. The smaller the confidence value, the less intuitive the *object* is. Therefore, only the *object* with the highest confidence value ($obj_{confid} > \alpha$) will apply the simplest and most straightforward description (Template 1), otherwise, we add the context of absolute position to help locate the *object* (Template 2). For the *object* whose text is not detected or recognized with low confidence, we leverage the context of its *neighbor* to help locate the target *object* (Template 3), e.g., tap the checkbox next to “Dark Mode”.

It is easy to describe a *SCROLL* action by its scrolling direction and offset (Template 5), e.g., scroll up a quarter of the screen. However, such an offset description is not precise and intuitive. To address this, if a new element with text appears by scrolling, we add this context to help describe where to scroll to (Template 4), e.g., scroll down half of the screen to “Advanced Setting”.

The description of *INPUT* is similar to *TAP*. For the high-confidence *object* with text (Template 6), it generates: Input [*text*] in the [$obj_{text}$] edittext. Different from the *TAP* descriptions, we do not apply the context of absolute position to help locate the low-confidence *object*. This is because the *objects* are gathering at the top when the keyboard pops up, so the absolute positioning may not help. Instead, we use the relative position of *neighbor* to describe the input *object* of which text is not detected or recognized with low confidence (Template 7), e.g., Input “John” in the edittext below “Name”.

After generating the natural language description for each action clip, we embed the description into the recording as subtitles as shown in Fig. 6. In detail, we create the subtitles by using the Wand image annotation library [42] and synchronize the subtitle display at the beginning of each action clip.