

## Appendix of “Mobile OS Task Procedure Extraction from YouTube”

### A Dataset Collection Process

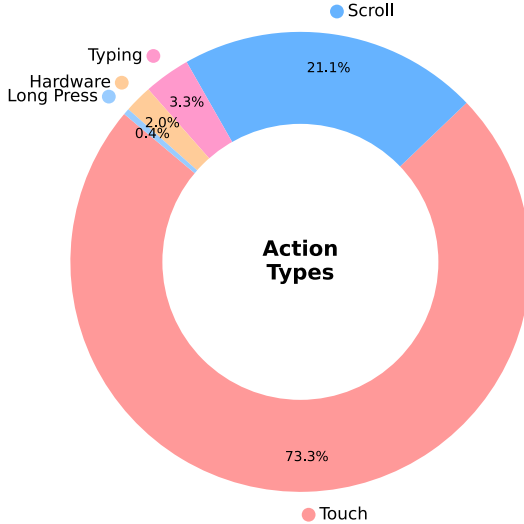


Figure A: Action distribution

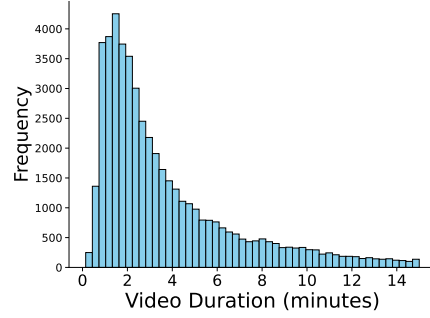


Figure B: Distribution of video duration

Mobile OS	Percentage
Android	42.87%
iOS	57.13%

Table A: Mobile OS distribution

Our dataset collection process leverages CommonCrawl web posts, specifically the C4 [22] and Dolma [23] datasets. We follow these steps:

We first filter posts using whitelisted domains, expanding beyond Android to include multiple mobile OS platforms. This approach, inspired by AndroidHowTo [24], ensures a diverse range of tasks across different operating systems.

Next, we employ GPT-3.5 Turbo Instruct [25] to identify and classify mobile OS-related tasks from the filtered posts. This step helps in extracting relevant task names.

Using the extracted task names as search keywords, we query YouTube for relevant instructional videos. Please refer to A.a for randomly sampled 45 examples of the search keywords. We download videos shorter than 15 minutes that have available transcripts, ensuring a manageable length for analysis while retaining narrative context. Please check Figure B for the video length distribution within 49K videos.

This process results in a diverse range of tasks that naturally follows the real-world distribution of mobile OS usage, reflecting actual user needs and interactions across various platforms. Current mobile OS distribution can be found in Table A. Not that we drop other mobile OS here as we were able to download less than 10 videos per those types.

Figure A presents the action distribution within the test dataset, where the most frequent action is touch (73.3%), followed by scroll (21.1%). Other actions such as typing, hardware interactions, and long-press constitute smaller portions in real-world videos. For our test data, please refer to the ‘hand\_annotation\_example’ in the Supplementary materials, and for the keywords used to download the test videos, see Section A.a in the same document.

## A.a List of Video Search Keywords

- How to add Dark Mode to Control Center on iOS
- How to cancel a subscription on iOS devices
- How to change the default browser on Xiaomi phones
- How to change your Instagram password on the web
- How to change your Spotify password on a browser
- How to change your Twitter password on mobile
- How to check battery usage on iPhone, iPad, or iPod touch
- How to clear cookies on iPhone Safari browser
- How to create a channel in Microsoft Teams mobile app
- How to deactivate your Viber account on Android
- How to delete Instagram search history on iOS and Android
- How to delete an Instagram comment on an iPhone
- How to delete an app on iOS 14
- How to delete your Google Search history
- How to download files from Google Drive on a computer or Android device
- How to enable and use Back Tap controls on iOS 14
- How to enable content protection in Telegram group or channel
- How to enable cookies on iPhone for Safari browser
- How to enable explicit lyrics filter on Spotify
- How to fix incorrect time on Google Authenticator app on Android
- How to fix low volume issues when using AirPods with an Android device
- How to free up storage space on your phone from the Telegram app
- How to hide a Viber chat on Android
- How to lock and unlock home screen layout on Samsung smartphones
- How to log out of Viber on iPhone or iPad
- How to make a playlist public on Spotify
- How to make your Twitter account private
- How to manage photo and video storage on iPhone, iPad, and iPod touch
- How to manage saved passwords in Chrome on Android
- How to measure someone's height using iPhone
- How to print from an iPhone or iPad using AirPrint
- How to react to messages on Telegram
- How to reset network settings on iOS or iPadOS device
- How to reset your WeChat password on Android
- How to restore a recently closed tab in Safari on iPad
- How to scan a QR code using the built-in camera on an iPhone or iPad
- How to sign out of Netflix on all devices at once
- How to sign out of iCloud on iPhone or iPad when the Sign Out button is greyed out
- How to spoof your location in Pokemon Go using Surfshark VPN
- How to turn off notifications for specific apps on iOS
- How to turn on dark mode for Facebook on iOS and Android
- How to turn on/off Wi-Fi and Bluetooth scanning on Android
- How to unblock someone on Viber on iPhone or iPad
- How to unsubscribe from mailing lists on a mobile device
- How to use Type to Siri on iPhone or iPad

## B Detailed Method Description

### B.a Scene Transition Detection

Our OCR-based scene transition detection algorithm operates as follows:

1. Extract text from consecutive frames in 4 frames per second (FPS) using Paddle OCR [15].
2. Compute the Levenshtein distance [18] between the text in an identical location but in adjacent frames.
3. Mark as transition if the distance exceeds 20% of the number of original text characters.

The threshold is determined empirically to balance sensitivity and robustness across different mobile OS interfaces.

## B.b Action Prediction

We first use GroundingDINO [14] for UI component area detection, followed by a post-processing step:

1. Apply GroundingDINO with a low confidence threshold (box threshold = 0.05, text threshold = 0.25, caption = “icon”) to detect potential UI elements from each frame.
2. Filter out unlikely candidates based on size and shape heuristics.
3. Merge overlapping bounding boxes to create a final set of interaction areas.

This process generates candidate touch regions used in the action prediction steps.

After detecting the UI component areas, we use the following prompts in our three-step VLM approach:

**Step 1:** “Summarize the visible UI elements and overall layout of this mobile screen.”

**Step 2:** “Based on the current frame summary and adjacent frame summaries, what action is likely being performed? Consider the potential interaction areas marked in the image.”

**Step 3:** “Given the predicted action and the zoomed image of potential interaction areas, select the most likely region for this action.”

The final action is determined by combining the predicted action type from Step 2 with the localization from Step 3. Please visit Section 3 for a detailed comparison with other approaches in action prediction.

## C Related Work

### C.a Mining Instructions from Videos

Early video understanding relied heavily on human-annotated datasets like CrossTask [26], TGIF-QA [27] and COIN [28]. However, the costly annotation process hindered their scalability. Consequently, researchers shifted focus to learning from unannotated videos, leading to datasets like VLOG [29] and HowTo100M [30]. While some works explored untrimmed task-specific videos, such as EpicKitchen [31] and Assembly101 [32], our focus lies in mining information from trimmed web videos about mobile OS navigation.

Recent advancements in video understanding have explored the use of instructional videos for learning task-specific knowledge [33, 34, 35]. Most of these approaches leverage the rich visual and textual information present in instructional videos to learn action sequences. However, most existing works focus on physical tasks in real-world environments, and the application of instructional video mining for mobile OS navigation remains largely unexplored.

### C.b Learning Task-Performing Agents

Initial web agents assumed access to HTML documents for next-action prediction, achieving reasonable performance without visual input [36, 37]. As the domain expanded and visual cues became essential, multimodal approaches combining vision and language emerged [38, 39, 40, 41]. This shift towards visual information has also influenced the development of mobile task agents, where modern mobile operating systems often restrict access to component information, necessitating image-based input. Consequently, AitW [9] has become a primary test-bed for mobile OS action planning [7, 8, 42, 10]. While these advancements are significant, the limited diversity and scalability of existing datasets hinder the development of genuinely generalized navigation agents.

Recent works have explored the use of reinforcement learning [43, 44, 45, 46] for mobile navigation, but their performance is often limited when generalizing to diverse real-world environments. In contrast to existing approaches, our work focuses on extracting mobile OS task procedure from the web. By leveraging the vast amount of user-generated content available online, we aim to overcome the limitations of manually annotated datasets and enable the development of more robust and generalizable navigation agents.

## D Episode Examples

Video search title: “How to make a playlist public on Spotify”

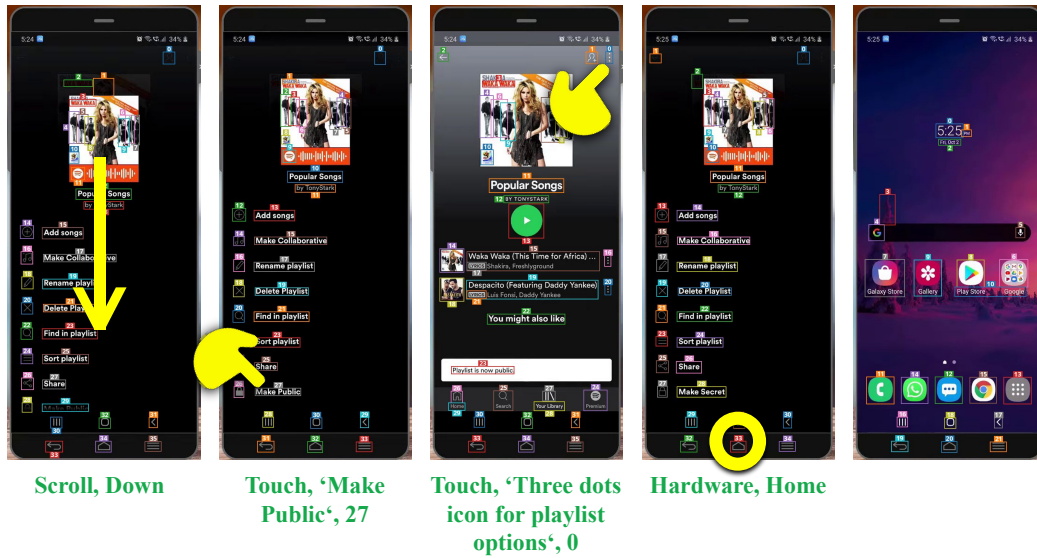
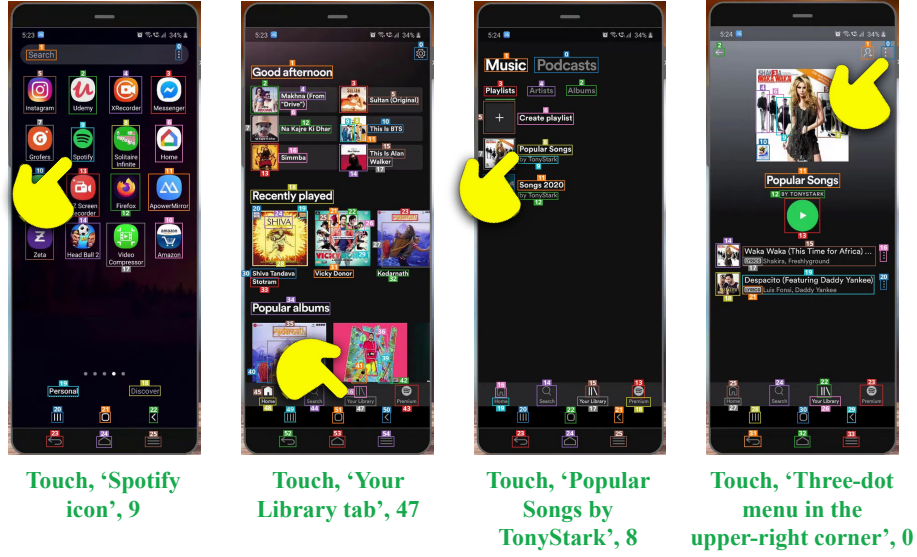


Figure C: An example episode in Android OS with action labels automatically annotated by MOTIFY. For each touch action, the box ID of the touch region is specified. **Green**: correct, **Red**: incorrect. A visual indicator is overlaid for a better understanding.

Video search title: “How to delete an app on iOS 14”

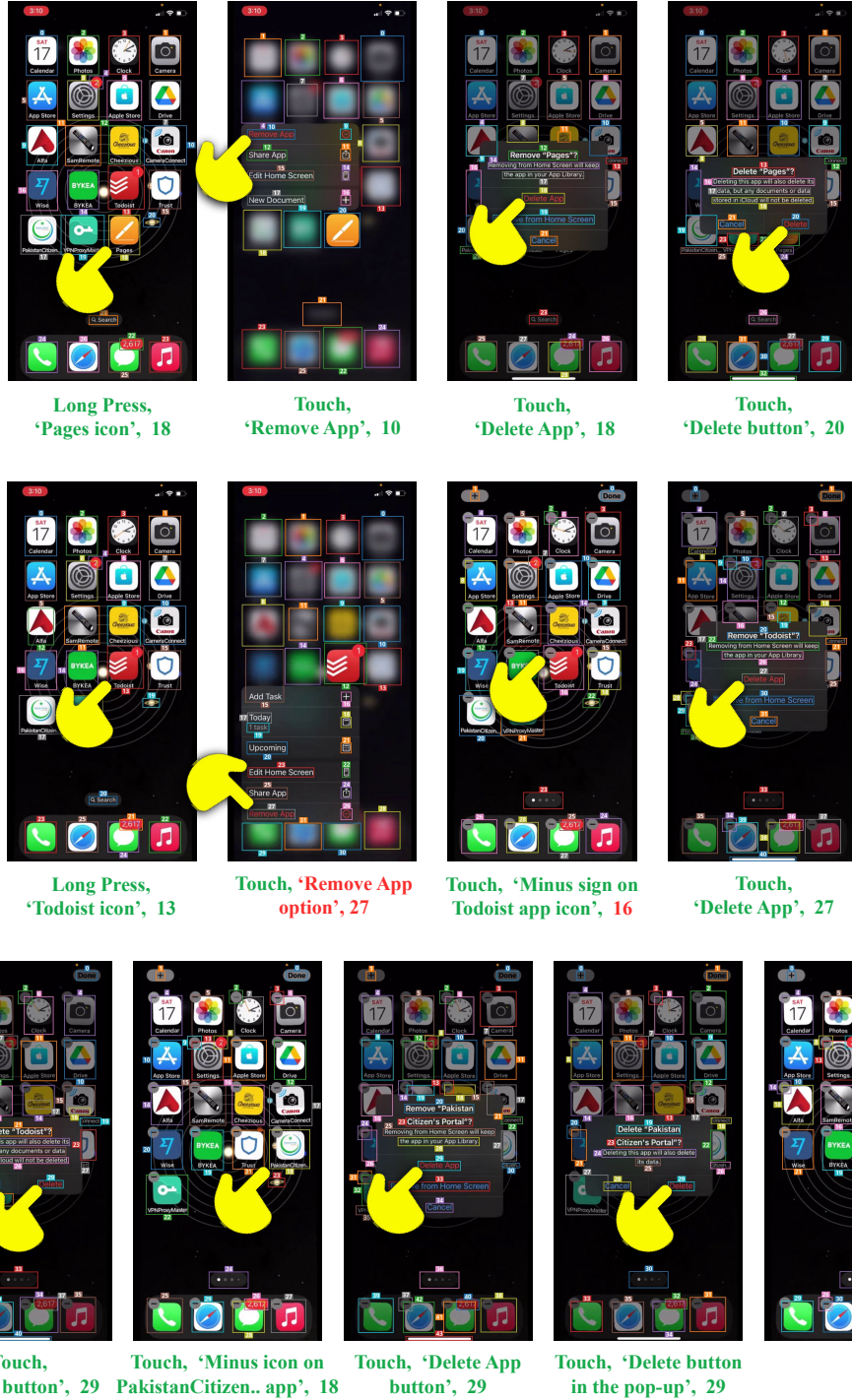


Figure D: An example episode in iOS with action labels automatically annotated by MOTIFY. For each touch and long press action, the box ID of the touch region is specified. **Green**: correct, **Red**: incorrect. A visual indicator is overlaid for a better understanding.