Mobile OS Task Procedure Extraction from YouTube

Yunseok Jang[*†], **Yeda Song**[*†], **Sungryull Sohn**[‡], **Lajanugen Logeswaran**[‡],
Tiange Luo[†], **Honglak Lee**[†‡]
[†]{yunseokj, yedasong, tiangel, honglak}@umich.edu
[‡]{srsohn, llajan, honglak}@lgresearch.ai

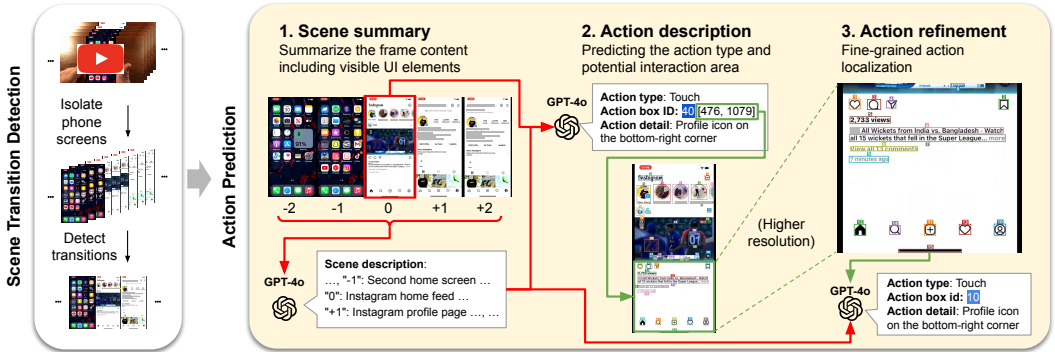


Figure 1: MOTIFY pipeline for mobile OS task inference from YouTube videos.

1 Introduction

Recent advancements in Large Language Models (LLMs) [1, 2, 3, 4] and Vision-Language Models (VLMs) [5, 6] have sparked significant interest in developing agents capable of mobile operating systems (mobile OS) [7, 8]. However, those mobile OS agents largely depend on datasets composed of a sequence of frames and corresponding actions, typically collected through manual processes or simulated environments [9, 10], which pose several challenges: time-consuming and costly data annotation, rapid OS updates rendering datasets quickly outdated, and limited diversity in user configurations and real-world scenarios.

Automatically extracting the actions and corresponding frames from online video could alleviate mobile OS agent data collection challenges, however recent efforts in this direction face limitations. While Android in the Wild (AitW) [9] supports mobile OS agent development, it is constrained by Pixel device emulators with system log messages. Video2Action [11] extracts scene transitions and actions from Android screencasts, but its single-OS focus and reliance on pixel-level differences make it unreliable in diverse real-world settings. Chen *et al.* [12] developed a method for predicting actions in mobile interfaces, but it still requires pre-training on a human-labeled Android dataset [13].

To address these limitations, we introduce Mobile OS Task Inference from YouTube, MOTIFY, a scalable approach for predicting actions and corresponding frames from mobile OS task videos with pre-trained VLMs. MOTIFY leverages publicly available YouTube content to capture real-world usage across diverse mobile OS platforms, eliminating the need for manual data annotation. Given any mobile OS task video, Mobile OS Task Inference from YouTube first detects scene transitions using GroundingDINO [14] and Paddle OCR [15], followed by action prediction with GPT-4o [16]. Leveraging these pre-trained VLMs eliminates the need for manual data-annotation process. Our experiments on both iOS and Android platforms show that our proposed pipeline outperforms SceneDetect [17] and YUV-diff [11] in scene detection accuracy, while also achieving the highest

[*]Equal contribution

Workshop on Video-Language Models at 38th Conference on Neural Information Processing Systems (NeurIPS 2024).

action prediction performance among all tested variants. Ultimately, MOTIFY provides a scalable and generalizable solution for mobile OS task data preparation, advancing mobile OS task understanding.

2 Method

Our goal is to automatically extract an agent's trajectory (*i.e.*, actions and corresponding frames similar to datasets like AitW [9]) for mobile OS tasks from instructional YouTube videos, without manual annotation or OS-specific simulators. Given a video, MOTIFY first identifies the frame where the agent took an action and then predicts the action performed on the identified frame. Note that, unlike AitW which uses system logs, our approach operates on real-world videos without system access. Since actions are not always directly visible in real-world videos, we detect transitions such as screen changes, button highlight, app launches, or menu shifts by comparing consecutive frames. These transitions serve as proxies for user input, allowing us to infer the action and its corresponding frame. Thus, our approach focuses on implementing two modules: scene transition detection (Section 2.1) and action prediction (Section 2.2). These components work together to transform raw video input into a structured, actionable format for mobile OS agent learning.

2.1 Scene Transition Detection

Unlike previous work that rely on vision-based features (*e.g.*, luminance difference in YUV) [11], we employ a novel Optical Character Recognition (OCR)-based approach for scene transition detection. This method proves more robust to varying interface configurations in real-world videos, as text rendering remains relatively consistent across different OS versions and user settings (*e.g.*, light/dark mode, recording conditions).

Our process begins with isolating the phone screen from the video using GroundingDINO [14] to handle distracting backgrounds in real-world videos. We then apply OCR using Paddle OCR [15] to extract text from consecutive frames. We detect scene transitions by computing the Levenshtein distance [18] between the detected text of adjacent frames and mark a transition if the distance exceeds a threshold. We construct a frame sequence for action prediction by selecting the middle frame between adjacent scene transitions.

2.2 Action Prediction

For action prediction, we first preprocess the frame to identify UI components. Given the lack of open-source models for icon detection in mobile interfaces [19, 20], we utilize GroundingDINO [14] to predict UI component areas, which are then used to generate a Set-of-Marks (SoM) style representation [21]. This approach allows us to precisely locate potential interaction areas without relying on platform-specific information. Please visit Supplementary Section D for visualization.

Given the preprocessed frame, we introduce a novel three-step approach for action prediction using GPT-4o [16]. First, we summarize the frames (without SoM representation) including visible UI components into a language description, providing a general understanding of the screen layout and available interactions without UI component occlusion. Next, we predict actions based on the summaries of current and adjacent (*i.e.*, two previous and two next) frames and the SoM representation of UI components, considering both temporal context and potential interaction areas. Finally, we refine the action prediction by feeding the zoomed image with SoM representation to precisely localize touch and long-press actions.

This multi-step process mitigates temporal misalignment issues and improves grounding accuracy, resulting in more precise and reliable action predictions across diverse mobile OS interfaces and tasks. By considering the current frame, adjacent frames, and potential interaction areas, our method can better understand and localize each action. Figure 1 illustrates our complete pipeline, showcasing how these components work together to extract mobile OS tasks from YouTube videos.

3 Experiments

To evaluate MOTIFY, we created a diverse test set from YouTube videos covering both iOS and Android platforms. We manually annotated these videos to create a ground truth dataset, identifying

scene transitions, actions, and bounding boxes for UI components. Our test set comprises 714 frames with 460 actions and 15,225 bounding boxes, reflecting the complexity of real-world mobile OS interactions (see Supplementary Section A.a for the search keywords to download the videos).

We first compared our OCR-based scene transition detection approach with an open-source scene cut algorithm [17] and the YUV-diff method used in Video2Action [11]. Table 1 shows that our method significantly outperforms these baselines, achieving an F1-score of 95.42%. This performance demonstrates the robustness of our approach across diverse mobile interfaces and video qualities.

Given the lack of code access for previous video-based mobile OS task extraction methods [11, 12], we conducted a comprehensive ablation study to evaluate our multi-step action prediction approach and the impact of action-focused summaries. Table 2 summarizes these results, demonstrating the superiority of our full method.

Our multi-image, 3-step approach consistently outperforms simpler variants, highlighting the importance of both temporal context and action-focused reasoning. The 2-step ablation omits step 3, while the 1-step ablation directly predicts the precise action from multi-image input, bypassing all intermediate steps. The performance drop from 3-step to 2-step (83.40% to 79.31%) demonstrates the value of our final refinement stage in precisely localizing actions. The more substantial decrease to 1-step (70.15%) underscores the complexity of mobile OS tasks and the necessity of multi-stage reasoning.

Notably, the "Multi-image w/o action" condition, which summarizes frames without emphasizing actions, shows a performance drop (81.66% vs. 83.40%). This suggests that action-oriented summaries provide crucial cues for accurate prediction. The single-image approach performs worst among first-step variants (76.96%), highlighting the importance of temporal context in mobile OS interactions.

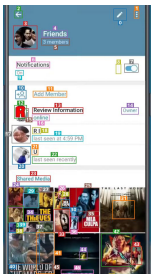
Figure 2 provides a qualitative comparison of action predictions from different configurations. This example illustrates how our 3-step approach with action-focused summaries can lead to more accurate and precisely localized predictions in a mobile OS scenario.

Table 1: Scene transition detection performance

Method	F1-score (%)
OCR-based (**Ours**)	**95.42**
SceneDetect [17]	81.00
YUV-diff [11]	71.00

Table 2: Action prediction accuracy (%)

Method	All	Touch
Multi-image 3-step (**Ours**)	**83.40**	**91.57**
Number of steps:		
2-step	79.31	86.83
1-step	70.15	73.69
First-step input:		
Multi-image w/o action	81.66	88.74
Single-image	76.96	87.86



Multi-image 3-step (Ours): Touch, 'Back arrow icon', box 2

2-step: Touch, 'Back arrow on the top left corner', box 3

1-step: Scroll, Down

Multi-image w/o action: Touch, 'Pencil icon', box 0

Single-image: Touch, 'Three dots menu icon', box 1

Figure 2: Qualitative comparison of action predictions. Green: correct, Red: incorrect.

4 Conclusion

MOTIFY demonstrates the potential for mobile OS task inference from real-world videos. Our method efficiently handles diverse mobile interfaces and user configurations, outperforming existing techniques in both scene transition detection and action prediction.

While effective, our current approach has some limitations. The reliance on GPT-4o [16] for action prediction, though powerful, could be addressed by exploring open-source alternatives tailored for mobile OS tasks. Additionally, while promising, the accuracy of our VLM-based extraction could be further improved through open-sourced specialized icon prediction models or fine-tuned VLMs.

Future work will focus on expanding our approach to a larger scale, aiming towards fully automatic data generation for mobile agent training. We are currently collecting a more extensive dataset using the method described in the supplementary material. This expanded dataset will enable further research into mobile OS navigation agents and video understanding tasks.

In conclusion, MOTIFY opens up new possibilities for mobile agent development and video understanding research, offering a scalable and adaptable approach to extract mobile OS tasks from real-world videos. We encourage the research community to build upon this work, leveraging its potential to advance the field of mobile OS task understanding and automation.

Acknowledgements

We thank Dong Ki Kim for helpful discussions. This work was supported in part by LG AI Research, OpenAI Researcher Access Program, and Kwanjeong Educational Foundation Scholarship.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 2019.

[2] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. *OpenAI Blog*, 2019.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *NeurIPS*, 2020.

[4] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training Language Models to Follow Instructions with Human Feedback. In *NeurIPS*, 2022.

[5] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. In *NeurIPS*, 2023.

[6] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved Baselines with Visual Instruction Tuning. *arXiv:2310.03744*, 2023.

[7] Zhuosheng Zhang and Aston Zhang. You Only Look at Screens: Multimodal Chain-of-Action Agents. *arXiv:2309.11436*, 2023.

[8] Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. SeeClick: Harnessing GUI Grounding for Advanced Visual GUI Agents. In *ACL*, 2024.

[9] Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy P Lillicrap. AndroidInTheWild: A Large-Scale Dataset For Android Device Control. In *NeurIPS Dataset*, 2023.

[10] An Yan, Zhengyuan Yang, Wanrong Zhu, Kevin Lin, Linjie Li, Jianfeng Wang, Jianwei Yang, Yiwu Zhong, Julian McAuley, Jianfeng Gao, et al. GPT-4V in Wonderland: Large Multimodal Models for Zero-Shot Smartphone GUI Navigation. *arXiv:2311.07562*, 2023.

[11] Sidong Feng, Chunyang Chen, and Zhenchang Xing. Video2Action: Reducing Human Interactions in Action Annotation of App Tutorial Videos. In *UIST*, 2023.

[12] Jieshan Chen, Amanda Swearngin, Jason Wu, Titus Barik, Jeffrey Nichols, and Xiaoyi Zhang. Extracting Replayable Interactions from Videos of Mobile App Usage. *arXiv:2207.04165*, 2022.

[13] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Ranjitha Kumar, and Jeffrey Nichols. Rico: A Mobile App Dataset for Building Data-Driven Design Applications. In *UIST*, 2017.

[14] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. In *ECCV*, 2024.

[15] Chenxia Li, Weiwei Liu, Ruoyu Guo, Xiaoting Yin, Kaitao Jiang, Yongkun Du, Yuning Du, Lingfeng Zhu, Baohua Lai, Xiaoguang Hu, Dianhai Yu, and Yanjun Ma. PP-OCRv3: More Attempts for the Improvement of Ultra Lightweight OCR System. *arXiv:2206.03001*, 2022.

[16] OpenAI. GPT-4o. https://platform.openai.com/docs/models/gpt-4o, 2024. Large Language Model.

[17] Brandon Castellano. *PySceneDetext*. https://www.scenedetect.com/, 2024. (accessed Sep., 2024).

[18] Vladimir Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. In *Soviet Physics Doklady*, 1996.

[19] Srinivas Sunkara, Maria Wang, Lijuan Liu, Gilles Baechler, Yu-Chung Hsiao, Jindong, Chen, Abhanshu Sharma, and James Stout. Towards Better Semantic Understanding of Mobile Interfaces. In *COLING*, 2022.

[20] Jieshan Chen, Amanda Swearngin, Jason Wu, Titus Barik, Jeffrey Nichols, and Xiaoyi Zhang. Towards Complete Icon Labeling in Mobile Applications. In *CHI*, 2022.

[21] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-Mark Prompting Unleashes Extraordinary Visual Grounding in GPT-4V, 2023.

[22] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *JMLR*, 2020.

[23] Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. *arXiv:2402.00159*, 2024.

[24] Yang Li, Jiacong He, Xin Zhou, Yuan Zhang, and Jason Baldridge. Mapping Natural Language Instructions to Mobile UI Action Sequences. In *ACL*, 2020.

[25] OpenAI. GPT-3.5 Instruct. https://platform.openai.com/docs/models/gpt-3-5, 2023. Large Language Model.

[26] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-Task Weakly Supervised Learning from Instructional Videos. In *CVPR*, 2019.

[27] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. TGIF-QA: Toward Spatio-Temporal Reasoning in Visual Question Answering. In *CVPR*, 2017.

[28] 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 *CVPR*, 2019.

[29] David F. Fouhey, Weicheng Kuo, Alexei A. Efros, and Jitendra Malik. From Lifestyle VLOGs to Everyday Interactions. In *CVPR*, 2018.

[30] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*, 2019.

[31] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100. *IJCV*, 2022.

[32] Fadime Sener, Dibyadip Chatterjee, Daniel Shelepov, Kun He, Dipika Singhania, Robert Wang, and Angela Yao. Assembly101: A Large-Scale Multi-View Video Dataset for Understanding Procedural Activities. In *CVPR*, 2022.

[33] Chuyi Shang, Emi Tran, Medhini Narasimhan, Sanjay Subramanian, Dan Klein, and Trevor Darrell. LUSE: Using LLMs for Unsupervised Step Extraction in Instructional Videos. In *ICCVW*, 2023.

[34] Yunseok Jang, Sungryull Sohn, Lajanugen Logeswaran, Tiange Luo, Moontae Lee, and Honglak Lee. Multimodal Subtask Graph Generation from Instructional Videos. In *ICLRW-MRL*, 2023.

[35] Lajanugen Logeswaran, Sungryull Sohn, Yunseok Jang, Moontae Lee, and Honglak Lee. Unsupervised Task Graph Generation from Instructional Video Transcripts. In *Findings of ACL*, 2023.

[36] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a Generalist Agent for the Web. In *NeurIPS*, 2024.

[37] Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. WebArena: A Realistic Web Environment for Building Autonomous Agents. In *ICLR*, 2023.

[38] Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating Multimodal Agents on Realistic Visual Web Tasks. In *ACL*, 2024.

[39] Jing Yu Koh, Stephen McAleer, Daniel Fried, and Ruslan Salakhutdinov. Tree Search for Language Model Agents. *arXiv:2407.01476*, 2024.

[40] Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. GPT-4V(ision) is a Generalist Web Agent, if Grounded. In *ICML*, 2024.

[41] Runliang Niu, Jindong Li, Shiqi Wang, Yali Fu, Xiyu Hu, Xueyuan Leng, He Kong, Yi Chang, and Qi Wang. ScreenAgent: A Vision Language Model-driven Computer Control Agent. *arXiv:2402.07945*, 2024.

[42] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. CogAgent: A Visual Language Model for GUI Agents. In *CVPR*, 2024.

[43] Daniel Toyama, Philippe Hamel, Anita Gergely, Gheorghe Comanici, Amelia Glaese, Zafarali Ahmed, Tyler Jackson, Shibl Mourad, and Doina Precup. AndroidEnv: A Reinforcement Learning Platform for Android. *arXiv:2105.13231*, 2021.

[44] Danyang Zhang, Hongshen Xu, Zihan Zhao, Lu Chen, Ruisheng Cao, and Kai Yu. Mobile-Env: Building Qualified Evaluation Benchmarks for LLM-GUI Interaction. *arXiv:2305.08144*, 2023.

[45] Juyong Lee, Taywon Min, Minyong An, Changyeon Kim, and Kimin Lee. Benchmarking Mobile Device Control Agents across Diverse Configurations. In *ICLRW*, 2024.

[46] Maayan Shvo, Zhiming Hu, Rodrigo Toro Icarte, Iqbal Mohomed, Allan D Jepson, and Sheila A McIlraith. AppBuddy: Learning to Accomplish Tasks in Mobile Apps via Reinforcement Learning. In *Canadian AI*, 2021.