# Detecting Keyframes in GUI Videos

Hao-Chun Shih, haochuns@umich.edu  Lei Lei, leilea@umich.edu  Pei-Chi Huang, peichi@umich.edu

Te-Hsiu Tsai, tedtsai@umich.edu  Yu-Hsin Huang, yuhsinhu@umich.edu

***Abstract*—This work focuses on automating keyframe segmentation in GUI (Graphical User Interface) videos, where actions involve tasks like clicking, scrolling, and typing rather than physical movements. Traditional action segmentation in real-world environments relies on human labeling, which is time-consuming and difficult to scale. Although several automated approaches exist for keyframe extraction, they typically underperform with GUI videos where changes between frames might be minimal. In this project, we address this problem by introducing transformers to extract features from GUI video. Additionally, we fine-tune state-of-the-art action segmentation models on the data. This will make large-scale GUI video datasets, such as WebArena, Mind2Web, and GUI-World, more accessible for video segmentation and downstream applications like tutorial understanding, video summarization, and training assistive GUI agents.**

***Keywords*—Action Segmentation, GUI Videos, Vision Transformer**

## I. Introduction

Action segmentation is a crucial task in video analysis, enabling the identification of meaningful events within a continuous video stream. While traditional action segmentation focuses on human activities in real-world environments, an emerging area of interest is GUI (Graphical User Interface) videos, where actions involve interactions with software interfaces rather than physical movements. By accurately detecting actions, this work can be extended to applications such as GUI tutorial understanding or summarization [1], and training assistive robots like GUI agents [2], [3], thereby enhancing automation in various domains.

In GUI videos, interactions typically include actions such as clicking, scrolling, dragging, and keyboard input, etc., and keyframes refer to the most representative frames within an action sequence. Identifying these keyframes can enhance the understandability of the content, allowing viewers to follow and perform tasks more effectively. Traditionally, this process has relied on human labeling, which is labor-intensive, costly, and difficult to scale. Additionally, manual annotations often suffer from inconsistencies, such as variations in marking the start and end points of keyframes. It is also a nontrivial task using simple gesture-matching techniques, such as those used in [4], due to the high dimensionality and inherent complexity of video streams.

This project aims to automate video annotation on GUI videos, providing more refined segment boundary detection and making large-scale datasets like WebArena, Mind2Web, and GUI-World more accessible for downstream applications. Particularly, we employ vision transformers (ViTs) for feature extraction in GUI videos. Unlike traditional feature extraction methods that rely on low-level pixel values or frame-wise convolutional neural networks (CNNs) like I3D [5], vision transformers utilize self-attention mechanisms self-attention mechanisms, which can capture more versatile changes within frames, thereby enhancing the interpretation of objects. Additionally, we imported and fine-tuned state-of-the-art (SOTA) methods to extract keyframes in GUI videos.

Our contributions are threefold:

- Utilizing vision transformers for feature extraction.
- Fine-tuning SOTA models for keyframe extraction in GUI videos.
- Constructing a large-scale annotated dataset to advance research in GUI videos.

## II. Methodology

In this project, we specifically address the challenges of segmenting GUI videos, which feature subtle changes between actions. Our approach focuses on two key aspects: 1) Enhanced video representation using ViTs and 2) Fine-tuned SOTA model on GUI videos.

### A. Video Feature Extraction

We introduce vision transformers (ViTs) to generate the video feature. Specifically, we use Swin Transformer [6], which is an improved version of Vit, designed to enhance computational efficiency and multi-scale feature learning for vision tasks. In the Action Segmentation task, actions in videos exhibit temporal variations and multi-scale features. Swin Transformer's hierarchical representation and shifted window mechanism allow it to capture characteristics more effectively. Unlike traditional ViTs, Swin Transformer uses local window attention to reduce computational complexity and its hierarchical feature enables the model to capture action details at different scales, improving the interpretation of subtle changes.

The output of Swin Transformer is a 6×6 matrix for each frame, which can be viewed as dividing the frame into several regions. Since the subsequent model expects one embedding per frame, we decide to use Spatial Attention to weigh each region's contribution (extracted from the output embedding of Swin Transformer), merging these regional features into a global embedding. This approach effectively captures the importance of different regions within the frame, allowing for the weighted fusion of regional features into a global representation. Additionally, it further helps the model identify which regions are most critical for final segmentation.

### *B. Action Segmentation Model*

In this project, we define actions in GUI videos based on user interactions involving mouse and keyboard inputs. We identified 7 types of interactions, as shown in the Table I:

TABLE I
ACTION TYPES

| **Device** | **Action** | **Device** | **Action** |
|---|---|---|---|
| **mouse** | scroll | **keyboard** | input |
| | hover | | delete |
| | drag | | enter |
| | click | | |

We implement state-of-the-art (SOTA) models models segment these actions in GUI videos. Specifically, we use:

- **Transformer-Based Models:** These models, such as FACT [7], ASFormer [8], and UVAST [9], were selected for their ability to capture long-term dependencies in video streams. These models are especially effective for refining frame boundaries and ensuring precise action segmentation.
- **Video Language Models (VLMs):** We also leverage advancements in VLMs, particularly SmolVLM [10], which is a compact Vision-Language Model designed for multimodal tasks. We applied a fine-tuned version of SmolVLM trained on the Mind2Web dataset for our video segmentation task.

We fine-tune and evaluate these models on our dataset and select the one with the best performance as our main method, continuously improving the model's prediction accuracy.

## III. RELATED WORK

### *A. Video Representation*

Traditional video segmentation relied on handcrafted features like optical flow and SIFT (Scale-Invariant Feature Transform) [11] which were effective for capturing local motion and key points but struggled to handle more complex variations and dynamic changes in the video. As videos became more complex and varied, deep learning techniques emerged, starting with Convolutional Neural Networks (CNNs) for spatial feature extraction and then 3D CNNs to capture temporal information [5]. RNNs like LSTMs and GRUs were also used for sequential modeling but faced efficiency and scalability challenges. These early methods laid the groundwork for modern transformer-based approaches.

Recently, transformer-based architectures, particularly vision transformers have been introduced for video feature extraction with enhanced interpretation of frame-wise information. ViT treats the image as a sequence of patches, and its transformer-based attention mechanism captures long-range dependencies and global context, making it effective for extracting rich visual representations. However, its quadratic complexity makes it computationally expensive, especially for high-resolution video tasks. Swin Transformer [6] addresses this issue by introducing hierarchical embeddings and a shifted window attention mechanism, reducing computational overhead while preserving spatial and temporal relationships. This is particularly advantageous for GUI video segmentation, where screen changes are often subtle. The shifted window approach in Swin Transformer allows it to effectively capture these small variations in the interface, making it more suitable for tasks that require fine-grained visual feature extraction.

### *B. Action Segmentation Model*

Early models used motion-based techniques like frame differencing and optical flow. With deep learning, CNNs were adapted for video segmentation but lacked temporal modeling. RNN-based approaches improved sequence learning but struggled with long-range dependencies. Temporal Convolutional Networks (TCNs) [12] later enabled efficient parallel processing across time, forming the basis for modern segmentation architectures. Current SOTA models have introduce transformer architecture for improved latent understanding in long videos. Originally designed for natural language processing, transformers leverage self-attention to capture dependencies across sequences. Unlike traditional dense models that perform frame-wise classification, transformers effectively model sequential video data with greater efficiency.

ASFormer [8] integrates multi-stage TCNs with transformers to reduce temporal complexity and efficiently segment actions. FACT [7] further refines segmentation by introducing cross-attention mechanisms that enable bidirectional information exchange between frames and actions. UVAST [9] leverages self-supervised learning, making it effective in both supervised and unsupervised segmentation tasks. BAFormer [13] explicitly models action boundaries to enhance temporal precision, preventing over-segmentation or under-segmentation issues. These models represent the cutting edge in video segmentation, addressing challenges in computational efficiency, accuracy, and boundary detection.

### *C. GUI Video Segmentation*

Based on our research, few segmentation models have been specifically fine-tuned for GUI videos. In Video2Action [4], they designed a segmentation model to identify and localize actions within videos by focusing on frame-to-frame changes and using topological methods for precise action localization. It detects subtle changes between frames, which is crucial for scenarios like GUI videos where actions involve minimal visual changes. By leveraging topological techniques, Video2Action improves its ability to understand spatial and temporal relationships, ensuring accurate segmentation of actions like button clicks, menu interactions, and scrolling. This approach enables the model to effectively handle the unique challenges of GUI videos, distinguishing it from general-purpose segmentation models that may struggle with such subtle and localized transitions.

### D. Our Approaches

Unlike conventional video segmentation models that are designed for natural videos with significant motion, our approach specifically targets GUI videos, where changes are often minor and localized. We incorporate transformer mechanisms to enhance segmentation accuracy, ensuring that even subtle interface changes are effectively modeled.

Traditional methods typically rely on direct frame inputs, such as raw RGB values or feature extraction through convolutional neural networks (CNNs). In contrast, our approach leverages transformer-based techniques to extract video features, improving the representation and interpretation of GUI elements.

Our segmentation model aligns with frameworks such as ASFormer [8] and FACT [7], which combine temporal convolutional networks (TCNs) and transformers to refine boundary detection and improve action segmentation. We adopt similar principles, aiming to improve segmentation precision in GUI videos by effectively modeling temporal structure and action transitions. Nevertheless, our segmentation model is tailored to capture fine-grained UI changes, which excels at detecting small yet meaningful updates like button clicks and menu interactions.

This GUI-specific adaptation differentiates our approach from general-purpose video segmentation techniques and enables more precise automation for UI analysis and testing.

## IV. Preliminary Experiment Results

### A. Dataset

GUI-world dataset encompasses various types of GUI videos, including IOS, Android, XR, websites and more. We decided to use the website subset of GUI-world, which contains a total of 2253 videos, as compared to IOS and Android interface, the website page is more element-detectable.

### B. Feature Extraction Settings

We extract frame images from videos at 30 FPS and resize the image to 192x192 pixels in order to fit the input dimension of Swin transformer. In our current embedding pipeline, we adopted the pretrained base swin2 model "swinv2_base_window12_192_22k" and configured the number of channels in spatial attention layer to 1024 to align with the output dimension of Swin transformer. Considering the limitation of memory, we only tested our embedding pipeline on the first 50 videos. With our current embedding pipeline, the storage and time required for embedding for a single batch (temporarily set as 25 videos per patch) are ˜200 MB and ˜10 mins. With a total of 2253 videos, the estimated storage and time to implement embedding are ˜2GB and 100 mins respectively.

To address the memory issue, we developed a script using Selenium to download and embed videos in patches to release memory in time. By using this script, all 2253 videos can be properly embedded in the future.

### C. Video Analysis

Figure 1 shows the distribution of duration for the first 500 videos. Most videos are shorter than 10s, which might indicate that tasks performed in videos are not particularly complicated and changes between frames can be trivial. Data augmentation may be necessary to overcome the inadequacy of context information.

### D. Action Analysis

We analyze the actions in all website videos. We defined an action as being composed of three elements: "mouse", "keyboard" and "input" and further categorized specific types for each of them to distinguish different actions, where "none" or "False" indicates no actions of this element. Figure 2 displays the distribution of deduplicated occurrences of action types for different elements, e.g., only counting "hover" once for a single video even if it occurs multiple times. Most actions are triggered only by the mouse, showing that the changes can be minor within a video.

We identified 18 distinct action combinations in the website GUI videos. Figure 4 shows the number of actions per video, with most videos containing 3-4 unique actions, and none exceeding 7 actions. However, the range of action changes within is wider, as shown in Figure 4, exceeding 6 changes in some cases, This suggests that certain videos exhibit frequent frame transitions and repetitive activities. Additionally, Figure 5 presents the distribution of the number of frames per action. Most actions span no more than 250 fps (˜8 seconds), which is relatively short compared to the typical video actions. Given these characteristics, our segmentation task must effectively capture both short-term and long-term patterns.

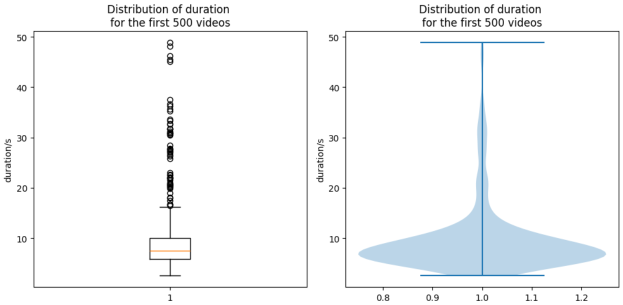

Fig. 1. Distribution of video durations.

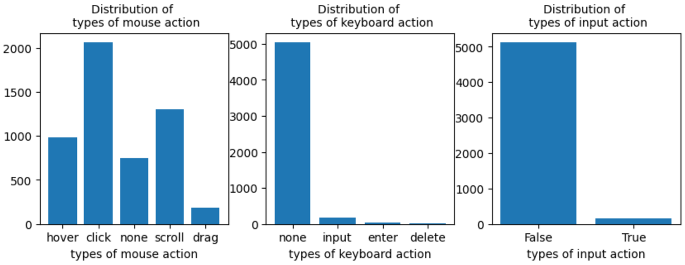

Fig. 2. Distribution of deduplicated occurrences (in a single video) of actions.

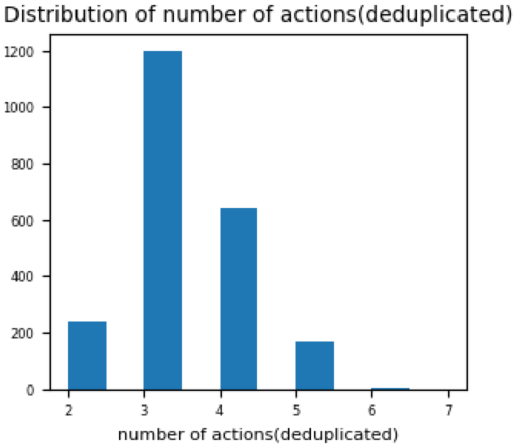

Fig. 3. Distribution of number of actions.

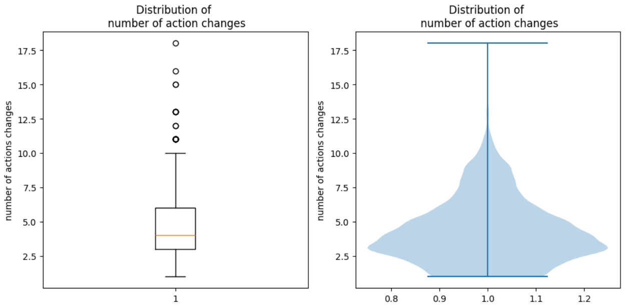

Fig. 4. Distribution of number of action changes.

### *E. Measurements*

We will evaluate our model performance with F1, R@1, frame-wise accuracy, and segmental edit distance. R@1 measures the ratio of correctly recalled steps, where a step is considered recalled if its most relevant video segment falls within the ground truth boundary. Frame-wise accuracy calculates the percentage of correctly labeled frames. Segment edit distance quantifies the cost of edit operations required to align segmentations.

## V. Future Milestones

Given current progress, we have identified four key directions for improving our performance:

**Fine-tuning SOTA models with GUI Data:** Currently, some of our experimental analyses involve models trained on non-GUI videos. While these models have achieved significant success in video segmentation, GUI-related tasks present a fundamentally different challenge, requiring a focus on distinct factors. Fine-tuning with our targeted dataset may provide a viable solution for improving performance in this domain. Jiang et al. (2023) [14] and Deng et al. [2] demonstrated the potential of fine-tuning existing methods for GUI tasks, showcasing promising results in adapting these models to new environments.

**Refined Model Architecture:** Customized models or architectures may also help improve the performance so far. The main architecture for our prediction is ViT embedding with powerful VLMs or temporal models. A more complex architecture designed specifically for GUI tasks or video segmentation may be feasible for our further progress. For example, Qian et al. [15] propose VideoStreaming model helps keep important memory for long video series.

**Boundary Segmentation Technique:** In our preliminary analysis, we find it challenging to accurately segment GUI videos using current methods. One key issue is the over-segmentation of the same action due to slight variations, leading to unnecessary splits. To improve segmentation consistency, exploring more robust boundary decision methods could be a potential direction for enhancement. Jiang et al. (2022) [16] introduce a novel approach to address this issue, known as the Importance Propagation Module. This module refines frame-wise segmentation by defining a scan range for each frame, computing similarity-based weighting factors, and applying context-aware importance adjustments. Through importance propagation, the model smooths out minor fluctuations in importance scores, preventing over-segmentation and improving key segment boundaries. By leveraging this method, we can achieve more stable and semantically meaningful video segmentation results.

**Resolution:** Due to the limited computation resource, we use only images with 192x192 resolution to do the embedding and modeling. However, resolution is crucial in GUI-related tasks because GUIs contain many small elements, including tiny icons, buttons, labels, and text that are difficult to recognize at low resolutions. In our case, it obviously leads to the problem of recognizing some GUI actions. Hong et al. [17] propose a solution for the dilemma by proposing a High-Resolution Cross-Module to address resolution limitations while maintaining computational efficiency. We can extract the embedded features for further prediction or analysis.

## VI. Schedule

Below is our weekly schedule:

TABLE II
Schedule

| Weeks | Goal |
|---|---|
| 3/16-3/22 | 1. Finish collecting data<br>2. Finish baseline model implementation |
| 3/23-3/29 | Implementing new features |
| 3/30-4/05 | Implementing new features |
| 4/06-4/12 | Training + fine-tuning 4/09 Exam Date |
| 4/13-4/19 | Evaluation and final adjustment |
| 4/20-4/26 | 1. Run on other possible datasets and evaluate<br>2. Prepare poster presentation |
| 4/27-5/03 | Prepare final report |

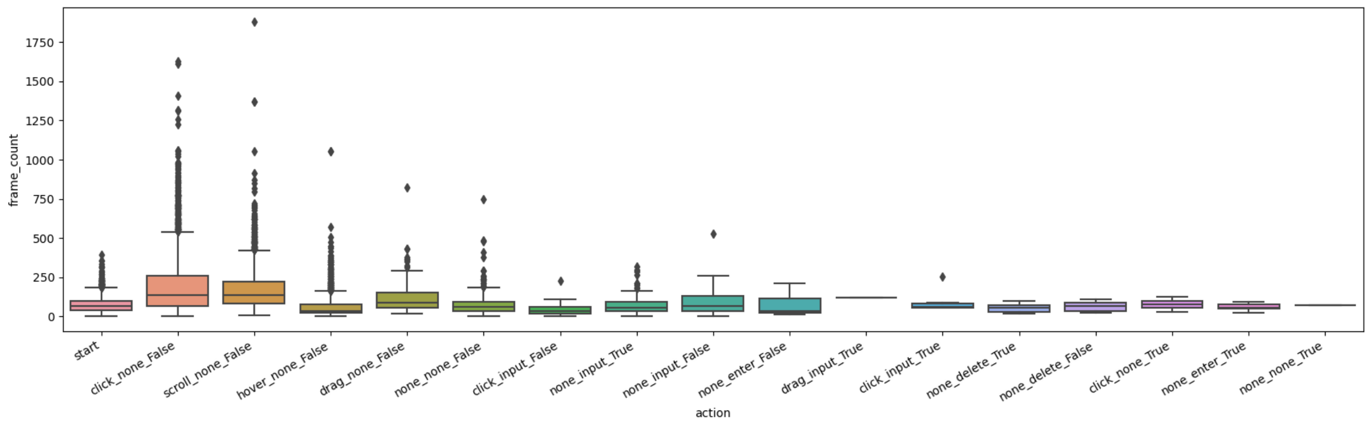

Fig. 5. Distribution of frame numbers per action.

## VII. Conclusion

We have established a pipeline for extracting video features from the GUI-World dataset and have successfully processed 50 website videos. Additionally, we have prepared the annotation files and are ready to integrate our dataset with the SOTA model. Future improvements include: 1) refining the ViT feature extraction process, e.g., improving the image resolution, for better frame understanding, 2) improving model architecture to address challenges specific to action segmentation in GUI videos 3) introducing boundary segmentation techniques to enhance segmentation accuracy and robustness. We will evaluate our model on the website videos and test its robustness to other types of GUI videos. Given that SOTA models typically achieve around 80% accuracy on general video action segmentation, we expect our approach to reach a similar level, making it competitive with existing benchmarks.

## VIII. Contributions

All the authors contributed equally to this project. Specifically,:

- **Lei Lei** and **Yu-Hsin Huang** focused on feature extraction from GUI videos.
- **Hao-Chun Shih**, **Pei-Chi Huang**, and **Te-Hsiu Tsai** worked on implementing the state-of-the-art (SOTA) models.
- All members actively participated in brainstorming, defining problem statements, conducting paper research, and writing the report.

## References

[1] T. Tuna, M. Joshi, V. Varghese, R. Deshpande, J. Subhlok, and R. Verma, "Topic based segmentation of classroom videos," in *2015 IEEE Frontiers in Education Conference (FIE)*, 2015, pp. 1–9.

[2] X. Deng, Y. Gu, B. Zheng, S. Chen, S. Stevens, B. Wang, H. Sun, and Y. Su, "Mind2web: Towards a generalist agent for the web," 2023. [Online]. Available: https://arxiv.org/abs/2306.06070

[3] S. Zhou, F. F. Xu, H. Zhu, X. Zhou, R. Lo, A. Sridhar, X. Cheng, T. Ou, Y. Bisk, D. Fried, U. Alon, and G. Neubig, "Webarena: A realistic web environment for building autonomous agents," 2024. [Online]. Available: https://arxiv.org/abs/2307.13854

[4] S. Feng, C. Chen, and Z. Xing, "Video2action: Reducing human interactions in action annotation of app tutorial videos," 2023. [Online]. Available: https://arxiv.org/abs/2308.03252

[5] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," 2018. [Online]. Available: https://arxiv.org/abs/1705.07750

[6] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9992–10 002.

[7] Z. Lu and E. Elhamifar, "Fact: Frame-action cross-attention temporal modeling for efficient action segmentation," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 18 175–18 185.

[8] F. Yi, H. Wen, and T. Jiang, "Asformer: Transformer for action segmentation," 2021. [Online]. Available: https://arxiv.org/abs/2110.08568

[9] N. Behrmann, S. A. Golestaneh, Z. Kolter, J. Gall, and M. Noroozi, "Unified fully and timestamp supervised temporal action segmentation via sequence to sequence translation," 2022. [Online]. Available: https://arxiv.org/abs/2209.00638

[10] A. Marafioti, O. Zohar, M. Farré, M. Noyan, E. Bakouch, P. Cuenca, C. Zakka, L. Ben Allal, A. Lozhkov, N. Tazi, V. Srivastav, J. Lochner, H. Larcher, M. Morlon, L. Tunstall, L. von Werra, and T. Wolf, "Smolvlm: Redefining small and efficient multimodal models," 2025.

[11] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157 vol.2.

[12] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks: A unified approach to action segmentation," 2016. [Online]. Available: https://arxiv.org/abs/1608.08242

[13] P. Wang, Y. Lin, E. Blasch, J. Wei, and H. Ling, "Efficient temporal action segmentation via boundary-aware query voting," 2024. [Online]. Available: https://arxiv.org/abs/2405.15995

[14] Y. Jiang, E. Schoop, A. Swearngin, and J. Nichols, "Iluvui: Instruction-tuned language-vision modeling of uis from machine conversations," 2023. [Online]. Available: https://arxiv.org/abs/2310.04869

[15] R. Qian, X. Dong, P. Zhang, Y. Zang, S. Ding, D. Lin, and J. Wang, "Streaming long video understanding with large language models," 2024. [Online]. Available: https://arxiv.org/abs/2405.16009

[16] H. Jiang and Y. Mu, "Joint video summarization and moment localization by cross-task sample transfer," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 16 367–16 377.

[17] W. Hong, W. Wang, Q. Lv, J. Xu, W. Yu, J. Ji, Y. Wang, Z. Wang, Y. Zhang, J. Li, B. Xu, Y. Dong, M. Ding, and J. Tang, "Cogagent: A visual language model for gui agents," 2024. [Online]. Available: https://arxiv.org/abs/2312.08914