Beyond Image Inclusion: A Position on Evidence-Centric Multimodality for Software Engineering Agents

Maleknaz Nayebi
York University
Toronto, Canada
mnayebi@yorku.ca

Xuchen Tan
York University
Toronto, Canada
swithin@yorku.ca

Abstract

Agentic software engineering is becoming multimodal. Benchmarks increasingly include screenshots, diagrams, and other visual artifacts, and agents increasingly advertise image support. Yet image presence does not imply that a task actually requires vision, and image ingestion does not imply meaningful multimodal reasoning. This position paper argues that the field should move from *image inclusion* to *evidence-centric multimodality*. Prior work shows that visual artifacts matter in software engineering practice, but recent rerun evidence on SWE-bench Multimodal suggests a multimodality gap: many issues originally solved with images remain solvable after image removal. We therefore argue for three shifts: benchmarks should establish *necessity*, not just presence; agents should adopt *selective modality use*, not default vision; and evaluation should measure *modality sensitivity*, not only end-task success. These shifts can make multimodal agent research in software engineering more diagnostic, efficient, and trustworthy.

1 The Problem: Image Presence Is Not Multimodal Evidence

Our daily communication is increasingly more visual and so is the development of the developers. They attach screenshots to issue reports, share interface captures in Q&A threads, and rely on diagrams and mockups to communicate states that are difficult to express in text alone [1, 3, 7]. This practical reality motivated the design of multimodal benchmarks such as SWE-bench Multimodal, which extends SWE-bench with issues containing screenshots, diagrams, and other visual artifacts [5, 10]. It also motivated multimodal agents and platforms, from generalist agent infrastructures such as OpenHands [9] to more specialized visual repair systems such as GUIRepair [4].

The field, however, risks making a category mistake. A benchmark that *contains* images is not necessarily a multimodal benchmark, just as an agent that *accepts* images is not necessarily using them. Recent work has already hinted at this tension. Nayebi shows that images are often meaningful and complementary in social coding, but does not claim that automated systems can use them reliably [7]. Ahmed et al. [1] find only modest gains from image processing for duplicate-question detection on Stack Overflow. More recent studies show that large language models can infer some intent from programming screenshots, but performance drops on cluttered or information-dense images and remains inconsistent overall [2, 3]. Images matter in practice but they are hard to use.

To ground our position, we draw on a controlled rerun analysis we conducted on SWE-bench Multimodal. We started with the subset of issues already solved by `OpenHands-Versa paired with Claude 4 Sonnet` as a multimodal agent in the original benchmark setting. We then reran those same issues under an otherwise identical setup, keeping the agent, model, repository environment, and evaluation pipeline fixed, but removing the associated images from the input. This isolates the contribution of the visual channel from the broader question of whether the agent can solve the issue at all. Out of 178 issues originally resolved with images, 148, or 83.15%, remained solvable without them. We then restored the original images for the 30 issues that failed in the text-only rerun and ran the agent once more under the same setup. This recovered only part of the lost performance: 12 issues became solvable again, while 18 remained unresolved. The implication is methodological rather than merely empirical. If most tasks remain solvable when the visual channel is removed under a controlled rerun, then benchmark scores may overestimate multimodal capability while actually reflecting stronger text-and-code reasoning.

2 Position I: Multimodal Benchmarks Should Require Images, Not Just Contain Them

Multimodal software engineering benchmarks should not merely collect issue threads that happen to include screenshots. They should establish whether the image provides the information necessary to solve the task. Broad image types are a weak predictor of necessity; what mattered was the information an image added beyond the issue report. Images showing desired output were often redundant, whereas images conveying steps, processes, or conceptual information were more likely to matter [3].

This observation aligns with prior work on software-related images. Eye of the Mind showed that images are frequently essential in human communication on social coding platforms [7]. ImageR sharpened that point by showing that screenshots improve bug-report clarity, but also that not all attachments help; roughly one in five images in its dataset failed to aid issue resolution [8]. The distinction is therefore not between "text" and "image" but between *evidence already recoverable from text* and *evidence that only the visual channel provides*. Benchmarks that ignore that distinction do not cleanly measure multimodal reasoning.

The practical consequence is clear. Future multimodal SE benchmarks should annotate images by *informational role*, not just artifact type. They should also verify *image–issue alignment*, to ensure that visual assets are relevant, traceable, and consistent with the task context. If an image is irrelevant or misaligned, the benchmark is no longer evaluating multimodal reasoning but noise tolerance.

3 Position II: Multimodal Agents Need Selective Vision, Not Default Vision

The same point applies to agents. Accepting images as input does not guarantee using them meaningfully. This is especially important in agentic software engineering, where multimodal processing is not free. Visual tokens increase cost and latency, and current multimodal agents already operate under complex tool-use and long-context constraints [6, 9]. Processing every available image by default is therefore neither efficient nor necessarily useful.

We argue that multimodal SE agents should move toward *selective modality use*. In such a design, the agent first reasons over issue text, repository context, and available execution signals, then decides whether the visual channel is likely to contain unique information. This decision can be guided by cues already identified in prior work. ImageR predicts when screenshots are beneficial in bug reports [8]. Studies of screenshot understanding distinguish between images that mainly show desired or observed output and images that encode deeper procedural or conceptual information [2, 3]. A selective agent should use such cues to decide when vision is likely to help and when it is likely to be redundant.

This point also suggests a different way to think about multimodal agent progress. Systems such as `GUIRepair` show that visual information can be incorporated effectively when the task truly depends on it [4]. But the right comparison is not "agent with vision vs. agent without vision". It is whether the agent uses vision *only when the image contributes task-critical evidence*.

4 Position III: Evaluation Must Measure Modality Sensitivity

If multimodal agents are to be evaluated rigorously, end-task success is not enough. We also need to know how sensitive performance is to the presence, absence, or degradation of visual input. The recent SWE-bench Multimodal rerun study began precisely from this point: an exploratory robustness analysis found that performance often remained stable even when images were degraded or removed, motivating the broader necessity analysis. That is the right instinct. Multimodal evaluation should include counterfactual tests.

We therefore argue that multimodal SE evaluation should report *modality sensitivity* alongside resolution rate. Benchmarks should support three checks: performance with the original image, performance with the image removed, and performance with the image restored or perturbed. These checks reveal whether the model relied on the image, whether the benchmark truly required vision, and whether the visual evidence was aligned and robust. Without such checks, multimodal scores conflate three different capabilities: extracting information from images, reasoning over information unique to the visual channel, and task resolving with text and code.

This recommendation also bridges current lines of work that otherwise remain disconnected. The screenshot studies by Ahmed et al. [2, 3] show what current models can and cannot infer from technical imagery. The negative results on duplicate detection show that simply adding image analysis can yield only marginal returns when the visual content is not the main bottleneck [1]. The benchmark work on SWE-bench and SWE-bench Multimodal demonstrates that agentic evaluation has shifted toward realistic issue resolution [5, 10]. Modality-sensitive evaluation connects it by asking the right comparative question: *when does vision change the outcome?*

5 Conclusion and Research Agenda

Agentic software engineering is becoming multimodal, but current evaluation still often mistakes image presence for multimodal evidence. Prior work shows that images matter in software practice, yet recent evidence shows that many benchmark tasks remain solvable without them [1–3, 7]. What matters is not whether an image is present, but whether it contributes extra information than text and code. Our position is therefore clear. Multimodal benchmarks should not be defined by image presence, but if they provide information necessary. Multimodal agents should not be evaluated solely by their ability to ingest visual input, but to determine when visual reasoning is actually required. Finally, multimodal evaluation should include explicit measures of modality sensitivity.

These observations point to a focused research agenda. Future benchmarks should be constructed so that textual and visual channels are deliberately complementary. Future agents should incorporate modality-selection policies that determine when visual processing is necessary. And future evaluation infrastructures should treat image necessity and image–task alignment as first-class benchmark metadata. The goal is not to maximize multimodality by default, but to design multimodality in a way that is diagnostically meaningful.

References

[1] Faiz Ahmed, Suprakash Datta, and Maleknaz Nayebi. 2024. Negative Results of Image Processing for Identifying Duplicate Questions on Stack Overflow. In *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. 281–291. doi:10.1145/3674805.3686688

[2] Faiz Ahmed, Xuchen Tan, Folajinmi Adewole, Suprakash Datta, and Maleknaz Nayebi. 2025. Inferring Questions from Programming Screenshots. *arXiv preprint arXiv:2504.18912* (2025). https://arxiv.org/abs/2504.18912

[3] Faiz Ahmed, Xuchen Tan, Folajinmi Adewole, Suprakash Datta, and Maleknaz Nayebi. 2026. Do Multimodal LLMs Understand Programming Screenshots? Inferring Questions and Extracting Relevant Content. *Empirical Software Engineering* 31, 88 (2026). doi:10.1007/s10664-026-10804-8

[4] Kai Huang, Jian Zhang, Xiaofei Xie, and Chunyang Chen. 2025. Seeing is Fixing: Cross-Modal Reasoning with Multimodal LLMs for Visual Software Issue Repair. In *2025 40th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 1156–1168.

[5] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R. Narasimhan. 2024. SWE-bench: Can Language Models Resolve Real-World GitHub Issues?. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=VTF8yNQM66

[6] Junwei Liu, Kaixin Wang, Yixuan Chen, Xin Peng, Zhenpeng Chen, Lingming Zhang, and Yiling Lou. 2024. Large Language Model-Based Agents for Software Engineering: A Survey. *arXiv preprint arXiv:2409.02977* (2024). https://arxiv.org/abs/2409.02977

[7] Maleknaz Nayebi. 2020. Eye of the Mind: Image Processing for Social Coding. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results*. 49–52. doi:10.1145/3377816.3381723

[8] Xuchen Tan, Deenu Yadav, Faiz Ahmed, and Maleknaz Nayebi. 2025. ImageR: Enhancing Bug Report Clarity by Screenshots. *arXiv preprint arXiv:2505.01925* (2025). https://arxiv.org/abs/2505.01925

[9] Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H. Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Yanjun Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, Junyang Lin, Robert Brennan, Hao Peng, Heng Ji, and Graham Neubig. 2025. OpenHands: An Open Platform for AI Software Developers as Generalist Agents. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=OJd3ayDDoF

[10] John Yang, Carlos E. Jimenez, Alex L. Zhang, Kilian Lieret, Joyce Yang, Xindi Wu, Ori Press, Niklas Muennighoff, Gabriel Synnaeve, Karthik R. Narasimhan, Diyi Yang, Sida I. Wang, and Ofir Press. 2025. SWE-bench Multimodal: Do AI Systems Generalize to Visual Software Domains?. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=riTiq3i21b