Position: Agentic Systems Constitute a Key Component of Next-Generation Intelligent Image Processing

Anonymous Author(s)
Affiliation
Address
email

Abstract

This position paper argues that the image processing community should broaden its focus from purely model-centric development to include agentic system design as an essential complementary paradigm. While deep learning has significantly advanced capabilities for specific image processing tasks, current approaches face critical limitations in generalization, adaptability, and real-world problem-solving flexibility. We propose that developing intelligent agentic systems, capable of dynamically selecting, combining, and optimizing existing image processing tools, represents the next evolutionary step for the field. Such systems would emulate human experts' ability to strategically orchestrate different tools to solve complex problems, overcoming the brittleness of monolithic models. The paper analyzes key limitations of model-centric paradigms, establishes design principles for agentic image processing systems, and outlines different capability levels for such agents.

1 Introduction

Image processing is a longstanding research area in computer vision. We have a wide variety of image processing and editing needs, ranging from post-photography editing, image restoration, enhancement, to style transfer. These tasks are inherently complex due to both the intricate nature of images and the unique aesthetic standards and nuanced expectations that humans hold. For a long time, image processing has been a specialized technical field managed by dedicated technicians and artists. Efforts in computer vision have long aimed to provide high-quality tools that enhance the efficiency and effectiveness of image processing tasks. *The research community strives to develop intelligent, adaptable software that maximizes convenience for users at all levels and fulfills a wide range of image processing needs.*

Figure 1: The existing research paradigm focuses on developing more powerful and multi-functional image processing models. In contrast, we advocate a new research paradigm centered on building agentic systems. Our goal is to create an agent that can integrate and leverage these models to achieve higher levels of intelligence, automation, and generality.

Early image processing algorithms were typically designed for specific types of problems, making them part of a broader pipeline or a standalone tool [5]. Professionals often need to configure and combine multiple processing steps to address particular image processing challenges. In the past decade, deep learning has driven a major leap in image processing, significantly improving the quality of individual tasks while introducing a more generalized, intelligent paradigm [14, 65]. The

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

industry has gradually shifted from constructing image processing pipelines to training end-to-end deep learning models that replace complex pipelines [3]. These deep learning models are now used not only to solve isolated problems but also to establish a general, multi-task, and intelligent solution that can operate effectively in diverse, real-world conditions [31, 50, 62, 68]. The advent of deep learning and artificial intelligence has made the vision of a general, intelligent, software-based "image processing assistant" seem closer than ever, though it remains just out of reach. **Significant research efforts have focused on the current paradigm, which is predominantly centered on developing various deep image processing models, as shown in Figure 1 (left).**

Nevertheless, the limitations of deep networks are emerging, and continuing within the existing research paradigm makes overcoming these constraints challenging. Firstly, these models face issues with generalization, as they perform well on test data similar to their training data but struggle on test data that deviates significantly from it [20]. Secondly, deep models capable of handling a wide range of degradation scenarios often compromise on quality and generalization [66]. Those that excel in specific degradations may lack generalizability, while models that handle a broad spectrum of tasks may not deliver peak performance on any single task [31, 67]. These challenges suggest that relying on a single model or fixed process for image processing may not effectively address the dynamic and complex real-world problems. Interestingly, despite the limitations of current image processing models, human artists and image editing professionals can still leverage these models and tools – often very simple ones, like basic operations in PhotoShop – to accomplish complex tasks that even the most advanced models cannot achieve. Emulating the dynamic and adaptive ways in which humans use these image processing tools could be a crucial step toward making image processing more intelligent and general. After all, no matter how powerful or multi-functional a tool may be, it still requires a capable operator – human or otherwise – to realize its true potential, see Figure 1 right.

In this position paper, we advocate for a new research paradigm centered on agentic-oriented image processing systems, offering a more autonomous, adaptable, and intelligent alternative to current methods. We begin by discussing the core capabilities required for intelligent image processing systems and the challenges that the current paradigm faces in Sec. 2. We then introduce the concept of AI Agents in Sec. 3, exploring their fundamental principles and role in intelligent systems. In Sec. 4, we extend this discussion to agentic image processing systems, analyzing how existing methods can incorporate varying degrees of agentic features to enhance generality and intelligence. We also examine the key characteristics of agentic image processing systems and outline different levels of agentic capability. Recognizing that large language models have become pivotal in the study of intelligent agents, we also explore how language models and multi-modal techniques may shape the future of image processing. Moreover, in Sec. 5, we highlight that there remains further room for exploration in certain critical attributes that determine a system's level of intelligence and generalization.

Alternative Views. The prevailing view holds that continued progress in developing models – through scale and improved architectures – could eventually overcome all the limitations and subsume the proposed agentic capabilities. While these views have merit, they underestimate the fundamental mismatch between static model architectures and the dynamic, compositional nature of real-world image processing requirements. Hybrid approaches combining foundation models with agentic components may offer a viable middle ground, but system intelligence requires explicit architectural support beyond current paradigms.

2 Backgrounds

2.1 Intelligent Image Processing

The core of intelligent image processing lies in developing intelligent and efficient algorithms that enable computers to automatically and accurately process images in various conditions to meet visual, psychological, or other needs. Ultimately, its goal is to build a "software employee" capable of automatically, intelligently, and effectively completing various image processing tasks. This is a highly visionary goal, and the community has long been approaching it from different angles in an attempt to simplify this challenging problem. Initially, image processing methods were mainly extensions of signal processing techniques applied to two-dimensional image signals, focusing on specific image processing operations. Entering the 21st century, with advancements in computing ability, many methods based on image priors and optimization have emerged but remain limited to

specific task scenarios. In the past years, the rise of machine learning and deep learning [32, 14] has propelled significant progress in the field of image processing. Particularly, the introduction of neural networks has led to breakthrough results in various image processing tasks. Data-driven methods not only allow for multiple image tasks to be handled within the same algorithmic framework but also make multitask integration and general-purpose image processing possible. Researchers have once again embraced the vision of intelligent image processing, and constructing a "software employee" capable of handling all tasks by unifying image processing tasks seems to have become a feasible direction. This position paper focuses on the core challenges of realizing this vision and the approaches to overcome them.

Specifically, an intelligent image processing system, a "software employee", needs to possess at least the following core capabilities:

- *Generality*: The system should be able to handle a wide range of diverse tasks without requiring separate models for each one, nor relying on extensive domain-specific training data or explicit task-specific instructions.
- *Autonomous*: The system should minimize reliance on user operations and supervision. It demonstrates proactivity by leveraging prior experience to explore new strategies without requiring explicit instructions.
- *Intelligence*: The system can adaptively adjust its processing strategies based on the semantic content and quality of the input, and user instructions. The system should demonstrate its intelligence and complexity in the processing workflow or in the final outcomes.
- *User Interaction and Feedback*: The system should facilitate clear, continuous, and user-friendly communication.
- *Self-Evolution and Creativity*: The system should generate meaningful and innovative solutions, going beyond straightforward problem-solving to provide novel approaches, insights, or outputs that showcase originality. Additionally, the system can continuously evolve and learn from new data, experiences, and user interactions.

2.2 Why Intelligent Image Processing is Challenging?

However, the current mainstream research paradigm, which centers around the development of deep learning models, struggles to align with the aforementioned vision.

The Challenge of Achieving Generality. Unlike high-level image understanding tasks, image processing tasks have both input and output as images that require precise pixel-level correspondence. The information needed for image processing tasks is not as specific as in image understanding. In image understanding, the model abstracts the image, extracts main features, and aligns them with semantics expressible in human terms. Although we hope that advanced deep networks for image processing can also learn the "semantics" of images, it is challenging to accurately describe local image details semantically. In fact, image processing networks do not learn semantics [19, 37, 22] but instead learn certain image transformations and overfit to training degradations [20, 39]. This is determined by the training paradigm of deep image processing models. Therefore, essentially, current image processing networks are not intelligent.

This leads to the next issue: the differences between various image processing tasks are also distinct from other types of tasks. Generally speaking, a specific image transformation or degradation can define an entirely new task. The differences between image transformations or degradations can be very subtle, and they can also be compounded to create almost unlimited types of transformations or degradations, resulting in virtually infinite image processing tasks. Due to deep models overfitting to the training set [8], tasks beyond the training scope cannot be well addressed [38]. This greatly limits the ability of current image processing methods to solve general problems. Worse yet, because collecting training images in the real world is extremely difficult, most research can only train on synthetic data, which further leads to generalization issues in practical applications.

Some methods attempt to include as many tasks as possible in the training set and train a sufficiently large model to achieve generality for common tasks [31, 11], even hoping that increasing the number of tasks will enable the network to generalize. However, these models have been proven to have a trade-off between the range of tasks and processing performance [66]. It's challenging to expand

the task range while keeping image processing performance from significantly declining. All these issues make constructing image processing systems with general capabilities highly challenging.

The Challenge of Developing Intelligence. Beyond the requirements of generality, we are increasingly emphasizing the intelligence these image processing systems exhibit. Firstly, we hope that image processing systems can explicitly perceive image content and perform targeted processing based on that content. For example, generating corresponding fur on animals or inferring and completing blurry or missing objects. Existing research indicates that end-to-end supervised deep image processing models do not possess this characteristic [37], but methods based on pre-trained generative models have demonstrated related capabilities and have thus achieved good results [62]. Secondly, we expect intelligent image processing systems to adaptively adjust processing strategies based on different input types or qualities, and even have the ability to make complex decisions based on specific image content. For instance, the system can automatically select the optimal denoising, enhancement, or restoration methods according to the image's resolution, lighting conditions, or noise levels. Additionally, we hope that image processing systems can dynamically understand users' complex needs. Currently, users need to select tools and set parameters based on their own expertise before obtaining results; this process does not reflect the system's intelligence. An intelligent system can accept user feedback or instructions to make dynamic adjustments in subsequent processing. These requirements have been mentioned to varying degrees in image processing research, but none have been explored in depth.

The Challenge of Balancing Autonomy, User Interaction, and Creativity. Existing approaches often fall into two extremes. On one hand, fully autonomous methods – such as end-to-end models – can quickly complete tasks but tend to overlook subtle user preferences, resulting in a rigid, one-size-fits-all automation. Automatic denoising may eliminate intentionally added artistic grain, and style transfer algorithms can homogenize diverse creative visions. Given the broad range and complex demands of image processing tasks, achieving consistently high-quality results proves challenging with these models. The end result is that people still need to pick and combine the results of different models, thus losing this automaticity. On the other hand, heavily manual interfaces impose a significant technical burden on users. Professional software like Photoshop requires extensive manual intervention and expert knowledge, which conflicts with the goals of ease of use and accessibility. Moreover, many existing approaches rely on single-model solutions with limited interactivity; more semantic, higher-level, and varied interaction mechanisms are needed to facilitate seamless communication between the user and the system.

Furthermore, existing methods also struggle to foster genuine creativity. Here, "creativity" goes beyond generating novel content via generative models [62]; it also involves discovering innovative ways to repurpose existing tools and deepening our understanding of them. As image processing evolves from mere technical correction into a creative medium, bridging this gap demands systems that not only "see" the pixels but also interpret the cultural, emotional, and contextual layers – a frontier that remains largely unexplored in current technology.

3 What is AI Agent?

An agent is a program designed to achieve its goals by perceiving the environment and interacting with it through available tools. These agents can operate autonomously without human intervention and proactively work towards their objectives [17, 34, 56]. From a design perspective, agent-based systems naturally fulfill our demand for automation. The various image processing models we develop can be regarded as tools of different scales and purposes, while the agent acts as the "coordinator" that actively orchestrates these tools, as shown in Figure 1 left. Early agent programs largely relied on symbolic methods [17] and reinforcement learning [24, 46, 63]. In recent years, however, agent systems powered by Large Language Models (LLMs) have achieved transformative progress [58, 7]. By training on massive text corpora through next-token prediction, LLMs demonstrate powerful knowledge transfer and logical reasoning abilities [4, 41, 1, 47, 25, 15], showcasing considerable potential in complex reasoning [51, 30], step-by-step planning [57, 58], and domain-specific knowledge applications [40, 21]. Compared to traditional reinforcement learning agents, LLM-based agents maintain long-term planning and simultaneously leverage broad general knowledge, thereby exhibiting more human-like cognitive characteristics [1]. From a cognitive standpoint, the fundamental responses of an LLM can be likened to "System 1," characterized by rapid, automatic thinking, whereas more advanced composite agent systems emulate "System 2,"

which involves deliberate, reflective reasoning [57, 35, 28]. Recent research has explored diverse agent architectures that enhance LLM-based problem-solving through structured mechanisms [43, 49] – such as tree- or graph-based search strategies [2, 57], external tool integration [44, 52], memory retrieval systems [70, 42], and error-driven learning processes [45, 58]. By combining an LLM's reasoning capabilities with structured problem-solving frameworks, these approaches show strong potential for tackling complex tasks [16].

Notably, pioneering efforts have employed LLM agents across various domains, all striving to create automated systems capable of proactively tackling a broad spectrum of challenges, aligning with the vision outlined in this position paper. For instance, frameworks like HuggingGPT [44] and Visual ChatGPT [44] leverage LLMs as multi-modal task controllers, integrating them with model libraries to decompose and solve diverse tasks; frameworks like OctoPack integrate LLMs with specialized toolsets, achieving significant performance gains in fields like medical image processing [40]. Advancements have also highlighted the effectiveness of LLM agents in tackling complex image processing tasks, achieving remarkable results [69, 9]. These advancements collectively highlight the transformative potential of LLM-based agents in addressing complex multi-modal challenges.

4 Agentic Image Processing System

The initial step toward agentic image processing involves acknowledging the fundamental reality that, **regardless of how advanced your image processing model is, carefully chosen preprocessing, postprocessing, or application-specific techniques/tricks can often enhance its performance**. For instance, certain severe degradations cannot be fully restored by a single pass through an image restoration model; applying the model iteratively to its own outputs can yield further improvements. Additionally, some degradations may lie beyond the training scope of the model, and introducing deliberate additional blurring before restoration can significantly mitigate these challenging cases. There exist numerous possibilities for such operations, and in practical applications, users frequently leverage these techniques to maximize performance.

4.1 Paradigms of Current Image Processing System

While this paper is the first to advocate for the construction of an agentic system to address challenges in intelligent image processing, traces of agentic thinking have already emerged, to varying degrees, in previous studies. We begin by examining the embodiment of agentic concepts behind the design of existing methods, adopting a perspective that progresses from simple to complex. Figure 3 provides a schematic illustration of these paradigms, offering a visual aid for better understanding.

End-to-End models are the most common paradigm in image processing research. Given an input, an end-to-end model produces a corresponding output. This category encompasses optimization-based, filter-based, and deep network models, with a focus on end-to-end deep network models for intelligent models. The standard approach involves collecting images that need processing along with their corresponding target images to form training image pairs, and then training the image processing model on this basis. This paradigm is the least agentic, and due to the following reasons, it has limitations in terms of generality and intelligence: Due to the limitations discussed in Sec. 2, no single model can simultaneously achieve both broad image processing capabilities and outstanding results. If a model is designed to be sufficiently "general," it will inevitably come at the cost of reduced performance on specific tasks.

Pipeline paradigm typically decomposes complex and difficult-to-model-at-once image processing problems into multiple independent processing steps. The main advantage of this approach is that it can effectively break down complex tasks into more manageable subtasks, allowing for the creation of new tasks through the combination of a limited number of image processing/operations [5, 3]. The modular design also equips the pipeline paradigm with high flexibility and scalability, enabling the system to be adjusted and updated according to specific needs. This makes it convenient to integrate new technologies or algorithms into the existing framework. For example, users can directly replace the denoising step with the latest denoising algorithm without redesigning the entire pipeline. Pipeline design is a typical idea of people to solve complex problems by combining simple tools.

Although pipeline models have advantages in handling complex tasks, their agentic level is still relatively low because each step is pre-defined based on practical applications and is difficult to adjust

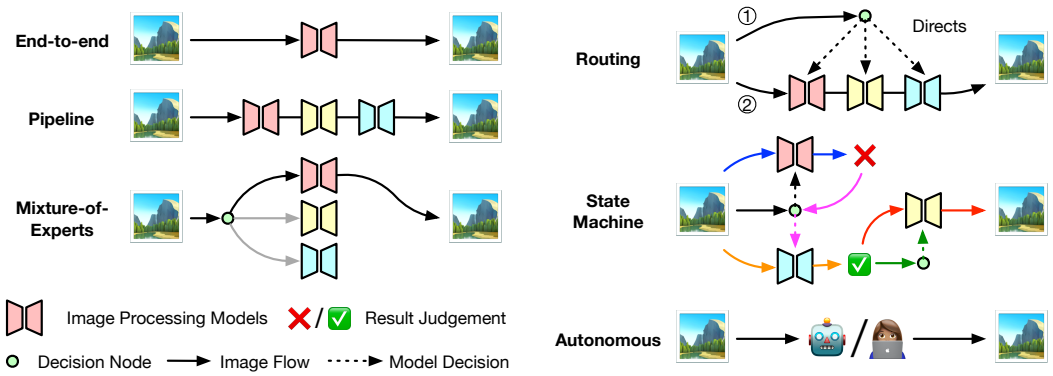

Figure 2: How image processing systems can embody different levels of agentic to enhance their generality and intelligence.

according to the diversity of inputs. Moreover, pipeline models often rely heavily on manual design, as the task decomposition and execution order within the pipeline significantly impact the results [63, 9, 69]. Therefore, they are used for specialized solutions to specific real-world problems rather than aiming for pursuing generality or higher levels of intelligence. However, by integrating different steps, the pipeline approach expands the application boundaries of image processing algorithms. This characteristic is a core advantage that agentic image processing systems can leverage.

Mixture-of-Experts (MoE) is another paradigm that broadens the task range of image processing systems and enhances performance by integrating the capabilities of multiple models. A single model is constrained by the trade-off between task coverage and processing effectiveness, making it difficult to efficiently handle each individual task while covering a large number of tasks. Similar phenomena have been observed in other large-scale model practices [6, 13, 12, 26]. To overcome these limitations, MoE typically introduces multiple expert models, each focusing on a specific task. The system dynamically selects the most suitable expert based on the input data or the task requirements, or combines the outputs of multiple expert models to optimize processing performance. This approach not only achieves a balance between task coverage and processing effectiveness but also allows for flexible adaptation to new tasks or improvement of performance on specific tasks by adjusting and replacing expert models. Therefore, MoE becomes an effective means to achieve task breadth while ensuring processing depth. Although MoE achieves a certain degree of agentic through proactive model selection, its generality and intelligence still depend on the performance of each expert model within the system. Since we cannot infinitely expand the number of expert systems, and there still exist problems that individual models cannot effectively solve, the generality of the MoE paradigm remains quite limited.

Routing is a combination of the Pipeline and MoE paradigms, potentially integrating the advantages of both. Similar to the MoE paradigm, the routing paradigm selects corresponding processing paths for input images to achieve targeted processing. However, unlike MoE, the routing paradigm selects a Pipeline composed of multiple models to maximally expand the range of feasible tasks. In essence, the routing paradigm automatically devises dedicated pipelines for different input image tasks and invokes the corresponding models. In other words, routing makes a "plan" for each input and executes it [23, 63]. The routing paradigm further enhances the system's agentic; when the decision-making methods are sufficiently accurate and robust, this paradigm can greatly expand the potential task coverage, thereby improving its generality. Since the decision-making process requires a deeper understanding of the images, the routing paradigm also possesses higher intelligence. However, once the path is determined, the outcome is already fixed. If an issue arises in an intermediate step, the routing approach cannot backtrack to address the problem at that specific step.

State Machine. Building on the foundation of the routing paradigm, state machines further expand to allow more fine-grained control over the processing flow. Similar to routing, a state machine produces a complex execution plan to conduct multiple image processing steps. However, due to the complexity of images and the variety of image processing operations, it is often not feasible to directly determine an optimal plan or parameter set in one run. In contrast to routing, the most notable feature of a state machine is its intelligent flow control: the system can reason and autonomously decide whether to proceed to the next step, adopt the current result or plan, or even undo the previous operation.

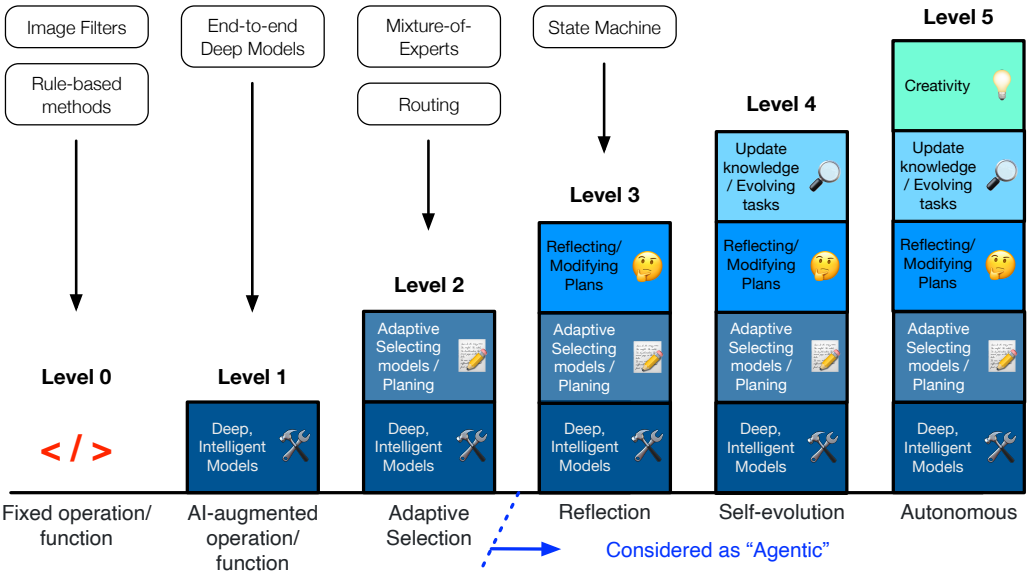

Figure 3: Levels of agentic capability in image processing systems, illustrating the progression from fixed rule-based methods (Level 0) to fully autonomous and creative systems (Level 5). Each level builds on the previous one, adding layers of adaptability, reflection, self-evolution, and creativity.

Essentially, the process by which humans solve problems can also be viewed as a highly flexible state machine. Some pioneering studies have already adopted this paradigm to build intelligent agent systems, demonstrating their remarkable intelligence and potential [69, 9, 36].

4.2 Characteristics of an Agentic System

Based on the above analysis, several core design principles of agentic thinking are already present in prior works to varying degrees. To further advance this approach, we summarize below the potential features of an agentic system – features that can significantly improve the system's intelligence, generality, and ease of use:

- *Proactive and Autonomous Problem-Solving*: An agentic system autonomously senses challenges, explores different models and methods, and dynamically adjusts strategies without relying on further human instructions. This allows for flexible and efficient image processing even in complex scenarios.
- *Integration of Multiple Models/Tools*: Rather than depending on a single model for complex image processing tasks, an agentic system can strategically combine multiple models or tools according to the specific task requirements or image characteristics. Even "all-in-one" large models can be combined with other operations or models to broaden coverage and improve performance.
- *Adaptive, Context-Aware Strategies*: An agentic system tailors its processing strategy based on the specific content or characteristics of the input image instead of applying a fixed pipeline. In other words, the system reasons about the image to make informed decisions.
- *Modular Architecture*: An agentic system often consist of multiple functional modules that work together – commonly including Perception, Reasoning, Action, and Reflection. Data and results flow through these modules, forming a coherent and synergistic workflow.
- *Easy Extensibility*: An agentic system should be readily extensible, allowing new features, tools or modules to be added without large-scale retraining or constructing massive new datasets. This flexibility enables the system to adapt to evolving requirements more effectively.
- *Continuous Reflection and Improvement*: An agentic system incorporates reflection mechanisms to evaluate and refine their performance. This iterative learning process ensures the system can leverage real-world usage data for sustained improvement over time.

4.3 Levels of Agentic in Image Processing

It is clear that "agentic" is not a binary concept but rather exists on a continuum. Drawing inspiration from the levels of autonomous driving [29], we propose a reference framework for classifying the agentic levels of image processing systems into six tiers, as shown in Figure 2. These levels reflect different characteristics that such systems may exhibit:

- **Level 0 (Fixed operation/function).** Methods at this level only provide basic, fixed transformations and processes. Regardless of the input image, they perform operations strictly according to predefined rules, such as filter-based or rule-based transformations. This stage exhibits almost no intelligence or generality.
- **Level 1 (AI-augmented operation/function).** While still focused on specific tasks, systems at this level go beyond simple rules by incorporating complex patterns learned through deep learning or other data-driven approaches. Although their performance surpasses Level 0, they remain limited in generalization.
- **Level 2 (Adaptive Selection).** Starting from this level, image processing systems no longer rely on a single model or tool. Instead, they can adaptively select and integrate different models, thus expanding the range of tasks they can handle. The ability to choose different processing strategies based on the input image demonstrates a certain degree of intelligence.
- **Level 3 (Reflection).** This level introduces more freedom in workflow control and the ability to reflect on output results, allowing systems to flexibly adjust strategies and processes. Through reflection and iterative adjustments, systems at this level can tackle a wider variety of problems.
- **Level 4 (Self-evolution).** At this level, agentic surpasses what fixed architectures can achieve. Systems can continuously learn and evolve from large amounts of data and experience, distilling new knowledge to solve previously unsolvable problems. They may even modify or advance their own workflows.
- **Level 5 (Fully Autonomous).** At the highest level, agents can execute image processing tasks autonomously without user intervention. In addition to incorporating all capabilities of the previous levels, they possess a degree of creativity, enabling innovative problem-solving approaches (e.g. discovering new tricks that the people who created the agent don't know about.). As a result, they can potentially replace human experts and approach a form of artificial general intelligence (AGI).

4.4 The Role of LLMs

Incorporating agentic design enables greater autonomy and adaptability. Current image processing paradigms can partially fulfill these objectives if designed with sufficient complexity (e.g., approaches based on reinforcement learning [23] or expert systems [64]). However, due to the inherent complexity of image processing tasks and the ambiguity of their descriptions, existing paradigms struggle to further develop and leverage agentic capabilities. LLMs have demonstrated strong adaptability when dealing with open-domain problems.

When an agent autonomously tackles complex tasks, it often faces numerous scenarios arising from the interplay of diverse factors. Given the complexity of image processing systems, these scenarios cannot be easily summarized or handled with simple rules. Traditional methods rely on predefined rules and features, which become insufficient in the face of combinatorial explosions. In contrast, LLMs can perform language-based reasoning and planning for each situation, offering remarkable generalization capabilities. Through this reasoning mechanism, abstract and unstructured demands can be mapped to specific image processing models or tools and translated into executable steps. Moreover, the workflow can be dynamically adjusted based on real-time feedback – for instance, rolling back or modifying the previous step. LLMs may even derive innovative new model combinations.

In a multi-modal setting, LLMs further provide the system with an "intelligent eye," enabling it to extract semantic information at multiple levels from abstract visual signals, far beyond what non-LLM approaches can achieve [61, 60, 53, 59]. Finally, natural language dialogue has proven to be an efficient and user-friendly channel for interaction. By employing LLMs, the system can engage in more flexible conversations with users, offer feedback, and accept instructions, thereby significantly enhancing both usability and extensibility.

5 Core Problems Demanding Further Study

Building intelligent agentic systems is a novel direction with many core challenges that require in-depth exploration. We analyze key issues that warrant attention in future research. Due to space constraints, we focus on two potential directions here and discuss additional topics in the appendix.

5.1 Cognitive Architecture of Image Processing

The foundation for building more complex agentic systems lies in designing their overall **cognitive architecture**[1]. A cognitive architecture refers to how the system "thinks" – in other words, the flow of code, prompts, and LLM calls that accept user input and execute operations or generate responses. Designing a cognitive architecture involves contemplating the abstract processes by which an intelligent agent solves problems at different levels. It's the methodology an agent uses to address a certain class of problems.

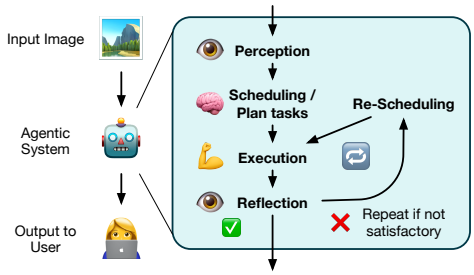

Figure 4: Cognitive architecture for image processing systems, illustrating the iterative process of perception, scheduling, execution, reflection, and rescheduling to achieve satisfactory results.

To facilitate understanding, we can start with the abstracted process of humans performing image processing or PhotoShop editing tasks. Figure 4 illustrates an example of a personified cognitive architecture for an image processing system, which is also the architecture used by Zhu et al. [69] and Chen et al. [9]. In this architecture, the interaction between the system and tools is abstracted into five stages: **Perception, Scheduling, Execution, Reflection, and Rescheduling**. Specifically, the Perception stage acts as the agent's "eyes," extracting necessary information from the input image. The Scheduling stage functions like the "brain," making judgments and formulating plans based on the acquired information and existing knowledge. The Execution stage represents "action," carrying out specific operations according to the plan. The Reflection stage evaluates whether the intermediate results meet expectations. If they do, the agent proceeds with subsequent plans; if not, the Rescheduling stage considers the failed results and modifies the original plan.

However, this intuitive cognitive architecture still leaves much room for improvement in the architectural research of image processing agent systems. For instance, for more specific problems, how should we design their cognitive architectures to meet the need for more refined control? For tasks requiring higher generality, how can we abstract a sufficiently general process to encompass a wider range of possible tasks? How can we systematically explore, distill, and abstract human problem-solving strategies into foundational principles for designing cognitive architectures? Furthermore, how can we create a cognitive architecture that surpasses human limitations, optimized specifically for intelligent image processing agent systems?

5.2 More Problems Demanding Further Study

Due to space limitations, we discuss four additional topics in the appendix: *Image Quality Assessment and Content Analysis*, *Knowledge Acquisition and Infusion*, *Human-Computer Interaction in Agentic Systems*, and *Exploitative Learning, Self-Evolution & Creativity*.

6 Conclusion

The evolution from task-specific models to agentic image processing systems marks a fundamental shift in addressing real-world complexity through dynamic tool orchestration rather than monolithic architectures. By embedding human-like adaptive reasoning into operational frameworks, such systems transcend current generalization limits while preserving specialized model strengths.

[1]The term "cognitive architecture" has a rich history in neuroscience and computational cognitive science. It refers both to theories about the structure of human thought and to computational implementations of these theories. Here, we borrow this concept but do not specifically refer to its original meaning.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690, 2024.

[3] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11036–11045, 2019.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] Mark Buckler, Suren Jayasuriya, and Adrian Sampson. Reconfiguring the imaging pipeline for computer vision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 975–984, 2017.

[6] Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. A survey on mixture of experts. *arXiv preprint arXiv:2407.06204*, 2024.

[7] Guangyao Chen, Siwei Dong, Yu Shu, Ge Zhang, Jaward Sesay, Börje F Karlsson, Jie Fu, and Yemin Shi. Autoagents: A framework for automatic agent generation. *arXiv preprint arXiv:2309.17288*, 2023.

[8] Haoyu Chen, Jinjin Gu, Yihao Liu, Salma Abdel Magid, Chao Dong, Qiong Wang, Hanspeter Pfister, and Lei Zhu. Masked image training for generalizable deep image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1692–1703, 2023.

[9] Haoyu Chen, Wenbo Li, Jinjin Gu, Jingjing Ren, Sixiang Chen, Tian Ye, Renjing Pei, Kaiwen Zhou, Fenglong Song, and Lei Zhu. Restoreagent: Autonomous image restoration agent via multimodal large language models. *Advances in Neural Information Processing Systems*, 2024.

[10] Haoyu Chen, Wenbo Li, Jinjin Gu, Jingjing Ren, Haoze Sun, Xueyi Zou, Zhensong Zhang, Youliang Yan, and Lei Zhu. Low-res leads the way: Improving generalization for super-resolution by self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25857–25867, 2024.

[11] Xiangyu Chen, Yihao Liu, Yuandong Pu, Wenlong Zhang, Jiantao Zhou, Yu Qiao, and Chao Dong. Learning a low-level vision generalist via visual task prompt. *arXiv preprint arXiv:2408.08601*, 2024.

[12] Zitian Chen, Yikang Shen, Mingyu Ding, Zhenfang Chen, Hengshuang Zhao, Erik G Learned-Miller, and Chuang Gan. Mod-squad: Designing mixtures of experts as modular multi-task learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11828–11837, 2023.

[13] Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

[14] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[15] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[16] Zane Durante, Qiuyuan Huang, Naoki Wake, Ran Gong, Jae Sung Park, Bidipta Sarkar, Rohan Taori, Yusuke Noda, Demetri Terzopoulos, Yejin Choi, et al. Agent ai: Surveying the horizons of multimodal interaction. *arXiv preprint arXiv:2401.03568*, 2024.

[17] Stan Franklin and Art Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. In *International workshop on agent theories, architectures, and languages*, pages 21–35. Springer, 1996.

[18] Jinjin Gu, Haoming Cai, Haoyu Chen, Xiaoxing Ye, Jimmy Ren, and Chao Dong. Image quality assessment for perceptual image restoration: A new dataset, benchmark and metric. *arXiv preprint arXiv:2011.15002*, 2020.

[19] Jinjin Gu and Chao Dong. Interpreting super-resolution networks with local attribution maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9199–9208, 2021.

[20] Jinjin Gu, Xianzheng Ma, Xiangtao Kong, Yu Qiao, and Chao Dong. Networks are slacking off: Understanding generalization problem in image deraining. *Advances in Neural Information Processing Systems*, 36, 2023.

[21] Rishi Hazra, Pedro Zuidberg Dos Martires, and Luc De Raedt. Saycanpay: Heuristic planning with large language models using learnable domain knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 20123–20133, 2024.

[22] Jinfan Hu, Jinjin Gu, Shiyao Yu, Fanghua Yu, Zheyuan Li, Zhiyuan You, Chaochao Lu, and Chao Dong. Interpreting low-level vision models with causal effect maps. *arXiv preprint arXiv:2407.19789*, 2024.

[23] Yuanming Hu, Hao He, Chenxi Xu, Baoyuan Wang, and Stephen Lin. Exposure: A white-box photo post-processing framework. *ACM Transactions on Graphics (TOG)*, 37(2):1–17, 2018.

[24] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.

[25] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[26] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[27] Gu Jinjin, Cai Haoming, Chen Haoyu, Ye Xiaoxing, Jimmy S Ren, and Dong Chao. Pipal: a large-scale image quality assessment dataset for perceptual image restoration. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 633–651. Springer, 2020.

[28] Daniel Kahneman. Thinking, fast and slow. *Farrar, Straus and Giroux*, 2011.

[29] Manzoor Ahmed Khan, Hesham El Sayed, Sumbal Malik, Talha Zia, Jalal Khan, Najla Alkaabi, and Henry Ignatious. Level-5 autonomous driving—are we there yet? a review of research literature. *ACM Computing Surveys (CSUR)*, 55(2):1–38, 2022.

[30] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[31] Xiangtao Kong, Jinjin Gu, Yihao Liu, Wenlong Zhang, Xiangyu Chen, Yu Qiao, and Chao Dong. A preliminary exploration towards general image restoration. *arXiv preprint arXiv:2408.15143*, 2024.

[32] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[33] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

[34] Chunyuan Li, Zhe Gan, Zhengyuan Yang, Jianwei Yang, Linjie Li, Lijuan Wang, Jianfeng Gao, et al. Multimodal foundation models: From specialists to general-purpose assistants. *Foundations and Trends® in Computer Graphics and Vision*, 16(1-2):1–214, 2024.

[35] Bill Yuchen Lin, Yicheng Fu, Karina Yang, Faeze Brahman, Shiyu Huang, Chandra Bhagavatula, Prithviraj Ammanabrolu, Yejin Choi, and Xiang Ren. Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks. *Advances in Neural Information Processing Systems*, 36, 2024.

[36] Yunlong Lin, Zixu Lin, Haoyu Chen, Panwang Pan, Chenxin Li, Sixiang Chen, Yeying Jin, Wenbo Li, and Xinghao Ding. Jarvisir: Elevating autonomous driving perception with intelligent image restoration. *arXiv preprint arXiv:2504.04158*, 2025.

[37] Yihao Liu, Anran Liu, Jinjin Gu, Zhipeng Zhang, Wenhao Wu, Yu Qiao, and Chao Dong. Discovering distinctive" semantics" in super-resolution networks. *arXiv preprint arXiv:2108.00406*, 2021.

[38] Yihao Liu, Hengyuan Zhao, Jinjin Gu, Yu Qiao, and Chao Dong. Evaluating the generalization ability of super-resolution networks. *IEEE Transactions on pattern analysis and machine intelligence*, 2023.

[39] Salma Abdel Magid, Zudi Lin, Donglai Wei, Yulun Zhang, Jinjin Gu, and Hanspeter Pfister. Texture-based error analysis for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2118–2127, 2022.

[40] Niklas Muennighoff, Qian Liu, Armel Zebaze, Qinkai Zheng, Binyuan Hui, Terry Yue Zhuo, Swayam Singh, Xiangru Tang, Leandro Von Werra, and Shayne Longpre. Octopack: Instruction tuning code large language models. *arXiv preprint arXiv:2308.07124*, 2023.

[41] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[42] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22, 2023.

[43] Tal Ridnik, Dedy Kredo, and Itamar Friedman. Code generation with alphacodium: From prompt engineering to flow engineering. *arXiv preprint arXiv:2401.08500*, 2024.

[44] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36, 2024.

[45] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[46] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[47] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[48] Haoran Wang, Yue Zhang, and Xiaosheng Yu. An overview of image caption generation methods. *Computational intelligence and neuroscience*, 2020(1):3062706, 2020.

[49] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024.

[50] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1905–1914, 2021.

[51] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[52] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023.

[53] Haoning Wu, Zicheng Zhang, Erli Zhang, Chaofeng Chen, Liang Liao, Annan Wang, Chunyi Li, Wenxiu Sun, Qiong Yan, Guangtao Zhai, et al. Q-bench: A benchmark for general-purpose foundation models on low-level vision. *arXiv preprint arXiv:2309.14181*, 2023.

[54] Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Yixuan Gao, Annan Wang, Erli Zhang, Wenxiu Sun, et al. Q-align: Teaching lmms for visual scoring via discrete text-defined levels. *arXiv preprint arXiv:2312.17090*, 2023.

[55] Haoning Wu, Hanwei Zhu, Zicheng Zhang, Erli Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Annan Wang, Wenxiu Sun, Qiong Yan, et al. Towards open-ended visual quality comparison. *arXiv preprint arXiv:2402.16641*, 2024.

[56] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*, 2023.

[57] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[58] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[59] Zhiyuan You, Xin Cai, Jinjin Gu, Tianfan Xue, and Chao Dong. Teaching large language models to regress accurate image quality scores using score distribution. *arXiv preprint arXiv:2501.11561*, 2025.

[60] Zhiyuan You, Jinjin Gu, Zheyuan Li, Xin Cai, Kaiwen Zhu, Tianfan Xue, and Chao Dong. Descriptive image quality assessment in the wild. *arXiv preprint arXiv:2405.18842*, 2024.

[61] Zhiyuan You, Zheyuan Li, Jinjin Gu, Zhenfei Yin, Tianfan Xue, and Chao Dong. Depicting beyond scores: Advancing image quality assessment through multi-modal language models. *arXiv preprint arXiv:2312.08962*, 2023.

[62] Fanghua Yu, Jinjin Gu, Zheyuan Li, Jinfan Hu, Xiangtao Kong, Xintao Wang, Jingwen He, Yu Qiao, and Chao Dong. Scaling up to excellence: Practicing model scaling for photo-realistic image restoration in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25669–25680, 2024.

[63] Ke Yu, Chao Dong, Liang Lin, and Chen Change Loy. Crafting a toolchain for image restoration by deep reinforcement learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2443–2452, 2018.

[64] MH Fazel Zarandi, Marzie Zarinbal, and Mina Izadi. Systematic image processing for diagnosing brain tumors: A type-ii fuzzy expert system approach. *Applied soft computing*, 11(1):285–294, 2011.

[65] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017.

[66] Ruofan Zhang, Jinjin Gu, Haoyu Chen, Chao Dong, Yulun Zhang, and Wenming Yang. Crafting training degradation distribution for the accuracy-generalization trade-off in real-world super-resolution. In *International conference on machine learning*, pages 41078–41091. PMLR, 2023.

[67] Wenlong Zhang, Xiaohui Li, Xiangyu Chen, Yu Qiao, Xiao-Ming Wu, and Chao Dong. Seal: A framework for systematic evaluation of real-world super-resolution. *arXiv preprint arXiv:2309.03020*, 2023.

[68] Wenlong Zhang, Guangyuan Shi, Yihao Liu, Chao Dong, and Xiao-Ming Wu. A closer look at blind super-resolution: Degradation models, baselines, and performance upper bounds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 527–536, 2022.

[69] Kaiwen Zhu, Jinjin Gu, Zhiyuan You, Yu Qiao, and Chao Dong. An intelligent agentic system for complex image restoration problems. *arXiv preprint arXiv:2410.17809*, 2024.

[70] Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, et al. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144*, 2023.

Appendix

A More Problems Demanding Further Study

A.1 Image Quality Assessment and Content Analysis

Regardless of how we design our cognitive architecture, certain fundamental capabilities are indispensable. Among these, the recognition, analysis, and evaluation of image content and quality are essential. In Figure 4, the Perception and Reflection stages are related to this capability; they

serve as the "eyes" of the image processing agentic system. Historically, image content recognition [48] and image quality assessment [27, 18] have been independent research fields, separate from image processing models, each with its own methods and objectives. However, from the research perspective of agentic systems, we impose higher demands on them.

For image content recognition, we need to consider its robustness under different image qualities and special circumstances. The recognized content must be designed with finer granularity to meet the specific needs of image processing, rather than being overly abstract like high-level vision tasks. Regarding image quality assessment, we cannot limit ourselves to evaluating a single "score." Models need to be more intelligent, performing fine-grained image quality analysis, determining types of distortion, and assessing the quality of intermediate results. The decisions of the entire agentic system largely depend on the accuracy and intelligence of this pair of "eyes."

Thanks to the development of multi-modal language models, a number of tools have now begun to demonstrate this capability [61, 60, 53, 54, 55]. Utilizing language models, these methods can describe image content, analyze image quality, evaluate the pros and cons of different image qualities, and provide judgments based on quality. These methods have already been applied in early research on agent-based image processing systems, showcasing their potential. However, the intelligence and accuracy of these methods still have a significant gap compared to large-scale practical applications.

A.2 Knowledge Acquisition and Infusion

After discussing the "eyes," let's turn to the "brain." As previously analyzed, the planning and decision-making abilities in intelligent agentic systems mainly stem from language models, which base their decisions on general knowledge learned from large volumes of text training data. Only when a language model has encountered problems and knowledge related to image processing during training can it be expected to make accurate judgments in the system; otherwise, the language model may struggle to provide reliable predictions. However, pre-trained language models usually contain only the most basic knowledge. If we want an image processing system to perform more specific and precise tasks, we need to supply the language model with the necessary knowledge. This involves two issues: the acquisition of knowledge and the injection of knowledge.

Firstly, acquiring such knowledge is non-trivial, and the method of injecting this knowledge into the system depends heavily on how it is represented. Image processing involves not only a large amount of conceptual and systematic knowledge but also a wealth of experience-based and case-based knowledge. This kind of knowledge is difficult to abstract into rules and usually exists in the form of case studies. Early attempts mainly employed two methods. Chen et al. [9] collected a series of input images along with corresponding instruction-output training data to implicitly carry a large amount of knowledge and information. The trained model then has the ability to handle similar problems. However, the drawbacks of this method are evident: firstly, collecting a large amount of high-quality training data requires substantial resources and is both costly and difficult. Secondly, the model lacks scalability; adding new knowledge requires retraining the model. Additionally, fine-tuning the language model may compromise its general capabilities.

In contrast, Zhu et al. [69] rely on the reasoning ability of an unmodified language model and provide a reference "manual." This method of supplying knowledge is known as Retrieval-Augmented Generation [33]. They first use the language model to summarize a large amount of scattered case information into knowledge that can be described linguistically. When solving actual problems, they provide the relevant content together. The language model utilizes its zero-shot learning and contextual inference capabilities to complete tasks based on the provided information. The advantage of this method is that it does not fine-tune the model, avoiding the loss of general performance, and the model still possesses strong reasoning and understanding abilities. However, its drawbacks lie in the difficulty of accurately describing professional knowledge in language. Moreover, quickly retrieving relevant information from a vast amount of knowledge is not easy, and it also places high demands on the design of the cognitive architecture.

In this area, we currently have only very preliminary results. The exploration, acquisition, representation, and injection of prior knowledge in image processing will become the core research topics of image processing agent systems.

A.3 Human-Computer Interaction in Agentic System

Agentic systems' multi-step operational paradigms, randomness, and natural language interfaces introduce new challenges in human-computer interaction. Currently, the primary way to interact with intelligent agent systems is through "chatting," where the system communicates its thoughts and actions in a conversational manner. However, we need new interaction methods to meet higher demands.

We must provide users with visibility into what the agent is doing by displaying all the steps it takes, allowing users to observe and understand the ongoing processes. Simultaneously, users should be able to give the agent more fine-grained and explicit instructions to control its behavior more precisely. Moreover, users should not only see what is happening but also have the ability to correct the agent. If they discover that the agent made an incorrect choice at step four (out of ten), they should be able to return to that step, correct the agent in some manner, and then proceed with the execution. The ultimate goal is to achieve collaboration between the agent and the user, enabling them to complete tasks together effectively.

A.4 Exploitative Learning, Self-Evolution & Creativity

The development of agentic systems has opened up new horizons in the fields of exploitative learning, self-evolution, and creativity. These concepts are crucial for advancing intelligent systems, enabling them to autonomously adapt, improve, and innovate over time without explicit human intervention, achieving higher levels of automation and agency as depicted in Figure 3.

Exploitative learning refers to the agent itself taking the initiative to determine the methods and content of knowledge acquisition within certain limits. The work of Chen et al. [10] embodies the prototype of this idea: they presented many experimental results to the agent, and the agent selected valuable content from them to learn. In some cases, the agent could even take some unconventional actions to acquire new knowledge through interaction with the world.

Self-evolution is the agent's ability to develop its own algorithms and strategies over time. This not only involves learning from data but also enables the agent to continuously improve itself based on its processing results, learning from past cases. Through iterative self-assessment and refinement, the agent gradually enhances its performance and may even modify its underlying processes to better adapt to changing environments or objectives.

Creativity in agent systems goes beyond mere problem-solving; it includes generating new ideas, methods, or outputs that are both original and valuable. This involves not only developing unique approaches to tackle complex image processing challenges that standard algorithms cannot handle, but also creatively generating content such as artistic transformations, stylizations, or entirely new visual effects.

These are grand visions under higher levels of agency and autonomy. At this stage, exploration of these issues is still quite limited. This paper serves only as an introduction to envisioning these higher levels of intelligence.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: The paper discusses the limitations of the work.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: No theory.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [NA]

   Justification: No experiments

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [NA]

   Justification: No experiments.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [NA]

   Justification: No experiments

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: No experiments

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [NA]

   Justification: No experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: tThe research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: tThere is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: tThe paper poses no such risks.

    Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We use LLM to proofreading.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.