

Orion: A Unified Visual Agent for Multimodal Perception, Advanced Visual Reasoning and Execution

VLM Run Research

We introduce **Orion**, a unified visual agent that integrates vision-based reasoning with tool-augmented execution to achieve powerful, precise, multi-step visual intelligence across images, video, and documents. Unlike traditional vision-language models that generate descriptive outputs, Orion orchestrates over a suite of specialized computer vision tools—including precise object detection, keypoint localization, panoptic segmentation, Optical Character Recognition (OCR), and geometric analysis—to execute complex multi-step visual workflows. The system achieves competitive performance across MMMU, MMBench, DocVQA, and MMLongBench while extending monolithic VLM capabilities to production-grade visual intelligence. Through its agentic tool-augmented approach, Orion enables autonomous visual reasoning that bridges neural perception with symbolic execution, marking the transition from passive visual understanding to active, tool-driven visual intelligence.

👉 Try Orion for free at chat.vlm.run • 📖 Learn more at vlm.run/orion

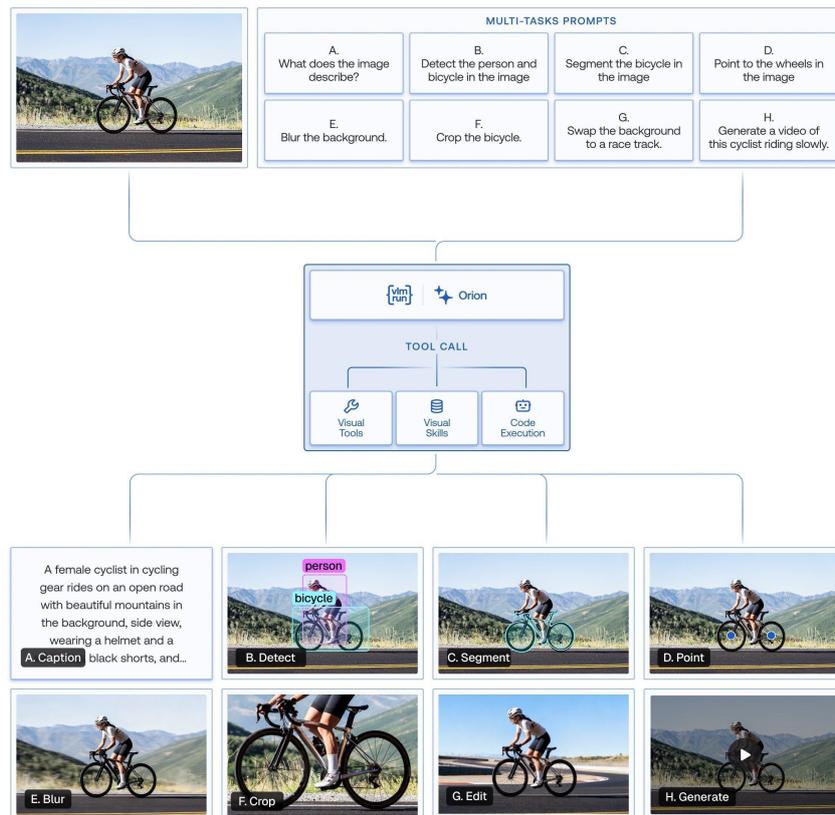


Figure 1 | **Orion's** Image Understanding, Reasoning, and Execution Capabilities: In the illustration above, we showcase 8 of the many capabilities of **Orion** - including captioning, detection, segmentation, pointing, tool-calling, image-generation and video-generation – all orchestrated from the original image as the input.

1. Introduction

Today’s frontier Vision-Language Models (VLMs)—GPT-5, Claude 4.5, Gemini 2.5 Pro (et al., 2025), and Qwen3-VL (Qwen Team, 2025)—can describe images, answer questions, and perform general-purpose reasoning over visual content. Building upon foundational vision-language models such as CLIP (Radford et al., 2021), LLaVA (Li et al., 2023a), and BLIP-2 (Li et al., 2023b), these systems have demonstrated remarkable capabilities in multimodal understanding. Yet they remain fundamentally limited: they operate as monolithic inference engines that generate descriptive or conversational outputs but cannot *act* on visual data with precision, determinism, or compositional control. They lack the fine-grained tool integration, multi-step planning, and structured validation required for production-grade visual workflows.

Orion is a visual agent framework that can take in any modality and generate any modality, using an agentic framework with multiple tool calling capabilities geared towards visual AI tasks to generate state-of-the-art results. The framework’s core innovation lies in its ability to seamlessly process and produce diverse data types—images, videos, documents, audio, and text—within a unified agentic architecture. This multi-modal flexibility enables **Orion** to handle complex visual reasoning tasks that require understanding, transformation, and generation across different data formats, from extracting structured information from documents to generating visual content from textual descriptions.

The system integrates the cognitive capabilities of large VLMs with the precision of tens of hyper-specialized computer vision tools—including Optical Character Recognition (OCR) for text extraction, detection for object identification, segmentation for pixel-level analysis, pointing for spatial localization, diffusion for image generation, and geometric analysis—orchestrated through an agentic architecture inspired by ReAct-style agent frameworks (Yao et al., 2023) and tool-augmented language models (Patil et al., 2023; Schick et al., 2023). The agentic framework enables **Orion** to dynamically reason about task requirements, select appropriate tools from its extensive library, and compose them into sophisticated multi-step workflows. Unlike baseline VLMs that generate single-step responses, **Orion** dynamically constructs and executes multi-step workflows from natural language instructions, achieving higher degree of task success through intelligent tool orchestration and iterative refinement.

Table 1 | Comprehensive capability comparison showing unique features of **Orion**. Unlike monolithic Large-Language Models, **Orion** delivers comprehensive capabilities across all modalities and tasks. **Specialized Skills** refers to tasks such as object localization, segmentation, image generation/editing, or geometric tools typically found in specialized computer vision applications. In the table below, the symbols represent the following:

✓ Full support, ✗ No support, ~ Partial support.

Modality	Task	Orion	GPT-5	Gemini 2.5	Claude Sonnet 4.5	Qwen3-VL 235B
Image / Video	Understanding	✓	✓	✓	✓	✓
	Reasoning	✓	✗	✗	✗	✓
	Structured Outputs	✓	✓	✓	✓	✓
	Tool-Calling	✓	✗	✗	✗	~
	Specialized Skills	✓	✗	~	~	✗
Document	Understanding	✓	✓	✓	✓	✓
	Reasoning	✓	✓	✓	✓	✗
	Structured Outputs	✓	✓	✓	✓	✓
	Tool-Calling	✓	~	~	~	✗
	Specialized Skills	✓	✓	~	✓	✗

The framework’s multiple tool calling capabilities enable it to tackle a wide spectrum of visual AI tasks that

were previously intractable for monolithic models. By combining specialized tools for document understanding, image analysis, video processing, and cross-modal reasoning, **Orion** can perform complex operations such as extracting structured data from invoices, analyzing medical imaging, tracking objects across video sequences, and generating contextually appropriate visual content. The agentic nature of the framework allows it to adapt its approach based on the specific requirements of each task, selecting the optimal sequence of tools and adjusting its strategy based on intermediate results.

This marks the transition from passive visual understanding to **autonomous, tool-augmented visual intelligence**—a new category of agentic AI platform capable of combining deep neural perception, symbolic reasoning, and structured automation within a unified system. **Orion** does not merely enhance existing multimodal models; it redefines what visual AI systems can achieve in complex, open-ended, and production-critical environments. The framework’s ability to generate state-of-the-art results across diverse visual tasks stems from its unique combination of large-scale neural reasoning, specialized tool precision, and intelligent orchestration, enabling it to outperform both general-purpose VLMs and task-specific systems.

Comprehensive evaluation across 46 diverse visual tasks—ranging from object detection and segmentation to complex multi-step visual reasoning, document analysis, and image generation—demonstrates that **Orion** consistently outperforms frontier VLMs including GPT-5 (Achiam et al., 2023), Claude 4.5, Gemini 2.5 Pro (et al., 2025), and Qwen3-VL (Qwen Team, 2025). Evaluations span established benchmarks including MMMU (Yue et al., 2024), MMBench (Liu et al., 2024), and DocVQA (Mathew et al., 2021). As shown in Figure 17, **Orion** achieves superior performance across all evaluated categories, establishing it as the leading system for production-grade visual tasks that require precision, multi-step reasoning, and specialized tool integration.

2. System Overview

2.1. System Architecture

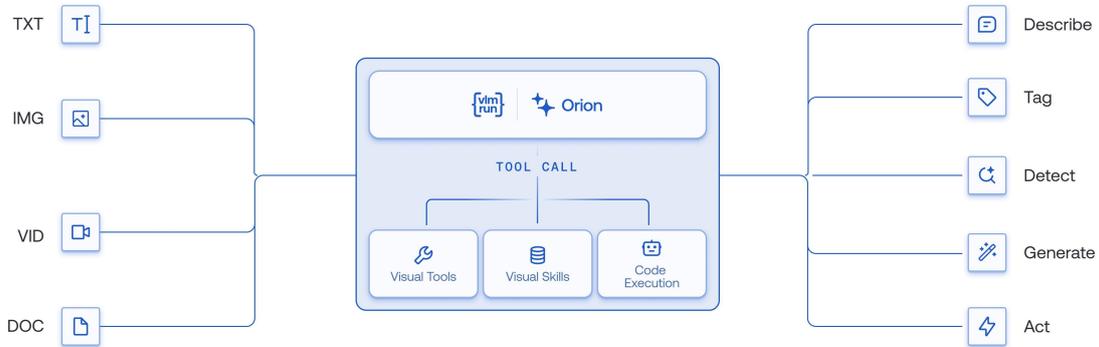


Figure 2 | System architecture of **Orion**, illustrating the visual agent and its interaction with various visual tools, skills and code execution environments. **Orion** supports text, image, video and document inputs to answer questions (in multiple steps if possible) that may involve captioning, tagging, detection, generation or external tool-calling. See capabilities 3 section for more details.

2.1.1. Unified Orchestration with Planning, Execution and Reflection

Orion implements a ReAct-style agent controller (Yao et al., 2023) that combines a large-language-model (LLM) based planner with a multi-modal tool-calling execution engine and a vision-language (VLM) based verifier. The core controller operates through three distinct phases (Plan-Execute-Reflect) in a closed-loop:

- **Plan** – **Orion** analyzes the user’s request to identify task intent and generates a structured execution plan, leveraging chain-of-thought reasoning (Wei et al., 2022) to decompose complex visual tasks. This planning phase determines the sequence of tool invocations, establishes data dependencies between operations, and evaluates the optimal choice of tools and execution order. The controller reasons about which visual processing capabilities are required and how they should be orchestrated to achieve the desired outcome.
- **Execute** – The planned sequence of tools is executed according to the established execution plan. Tools are invoked either sequentially or in parallel, depending on dependency constraints and computational cost considerations. This flexible execution model allows **Orion** to maximize throughput while respecting data dependencies between operations. The execution engine integrates with multi-modal tool-calling capabilities including GPU-accelerated tool invocation to enable complex, multi-step visual reasoning and structured output validation.
- **Reflect** – Intermediate and final outputs are evaluated against expected types and values to ensure correctness and consistency. This reflection phase identifies potential inconsistencies, validates visual output quality with respect to the inputs, and determines whether to retry failed operations, refine intermediate results, or finalize the completion request. The reflection mechanism enables **Orion** to course-correct and iteratively improve output quality throughout the execution process.

2.1.2. Multi-Modal Tool Execution

Orion provides first-class visual reasoning and execution capabilities through a comprehensive suite of specialized tools organized into four functional categories. The tool library comprises a suite of distinct tools spanning document understanding, image analysis, video processing, and cross-modal reasoning tasks.

Image Tools – These tools include object detection for identifying and localizing visual entities, face recognition for identity verification, segmentation for pixel-level classification, and visual question answering for image-based reasoning tasks.

Document Tools – These tools include Optical Character Recognition (OCR) for text extraction, layout detection for structural analysis, table extraction for tabular data parsing, form understanding for structured document processing, and keyword search for content retrieval within documents.

Video Tools – These tools include long-form visual transcription for video-to-text captioning, scene segmentation for temporal boundary detection, object tracking for motion analysis, and action recognition for activity classification.

Mixed-modality Tools – This category encompasses cross-modal retrieval for finding related content across modalities, visual redaction for privacy-preserving content masking, and region-level content extraction for targeted information retrieval.

Each tool exposes standardized input/output interfaces that enable the controller to interpret and react to tool outputs dynamically. This standardization allows interleaved reasoning and re-interpretation of visual outputs, incorporating results from one tool as inputs to subsequent execution steps. The uniform interface design ensures composability and enables the controller to construct complex multi-step workflows without manual intervention.

2.1.3. Multi-Modal Context and Session Management

The **Orion** system handles diverse input modalities—including images, videos, documents, audio, and text, making it especially challenging to manage context without loss of information within the agent’s context window. The system implements adaptive context management that selectively retrieves only the context relevant to the current instruction, rather than maintaining the entire conversation history. This approach enables efficient processing in multi-turn conversations extending beyond 25 turns while maintaining coherent session state across complex, long-running interactions.

2.1.4. Reflection with Visual Judges

The **Orion** system employs VLM-as-a-Judge (Lee et al., 2024) models to visually evaluate tool outputs against expected schemas, identify potential inconsistencies, and determine whether to retry, refine, or finalize the workflow. This reflection mechanism enables closed-loop control over the agent’s execution ensuring that intermediate and final results meet structural and semantic requirements of the original instruction.

Conditional Execution & Closed Loop Control: The reflection framework enables conditional invocation of VLM-as-a-Judge tools to evaluate the outputs of other tools within the execution plan. By incorporating these evaluation checkpoints, the system iteratively refines outputs and course-corrects the overall visual reasoning and execution trajectory. This feedback loop allows the controller to detect errors early, trigger corrective actions, and progressively improve result quality without manual intervention, enhancing both robustness and reliability across complex multi-step workflows.

2.2. Key System Features

Orion provides several distinctive capabilities enabling robust, flexible, and scalable multi-modal reasoning and execution in a single unified API.

Natively Multi-Modal – **Orion** is truly multi-modal, with native support for handling images, videos, documents, audio, and text within the same conversation. All modalities are processed consistently, enabling seamless cross-modal reasoning and execution without requiring separate workflows or manual format conversions.

Dynamic Routing – **Orion** implements dynamic routing of complex instructions and tasks to more sophisticated models and tools as needed. When the controller detects that a task exceeds the current agent’s capabilities, it can dynamically escalate to more powerful models or specialized tools, with graceful fallbacks ensuring robustness when preferred resources are unavailable.

Dual-Mode Agent Design – **Orion** supports a *dual-mode* agent design, operating in both *chat* and *structured-output* modes to serve conversational and API-driven use cases. Chat mode provides natural-language interaction for exploratory workflows (e.g., visual chat - <https://chat.vlm.run>), while structured mode delivers type-safe, schema-validated outputs for deterministic execution with high reliability.

Closed-loop Reflection – Orion incorporates closed-loop reflection to improve output quality and correctness. By evaluating intermediate and final results against expected schemas and semantic requirements, it iteratively refines outputs, detects errors early, and course-corrects execution trajectories without manual intervention.

2.3. Key API Features

Orion exposes a comprehensive API surface designed to support diverse integration patterns and use cases, from interactive conversational interfaces to programmatic batch processing pipelines. See our docs page for more details: <https://docs.vlm.run/agents/introduction>.

Multi-Modal Input Support – The API provides native support for passing images, videos, documents, audio, and text within the same conversation. Content can be provided via file identifiers or direct URLs, enabling flexible integration with existing storage systems and content delivery networks. This unified input mechanism eliminates separate endpoints or pre-processing steps for different modalities.

```
1 from vlmrun.client import VLMLRun
2
3 # Initialize the client
4 client = VLMLRun(base_url="https://agent.vlm.run/v1", api_key="...")
5
6 # Example: Chat completion with multi-modal inputs
7 image_file = client.files.upload("logo.jpg")
8 video_file = client.files.upload("racing.mp4")
9 messages = [
10     {"role": "user", "content": [
11         {"type": "text", "text": "Place this logo in the racing ad video?"},
12         {"type": "input_file", "file_id": image_file.id},
13         {"type": "input_file", "file_id": video_file.id}
14     ]}
15 ]
16
17 # Perform chat completion
18 chat_completion = client.agent.completions.create(
19     model="vlmrun-orion-1:auto",
20     messages=messages,
21 )
22 print(chat_completion.choices[0].message.content)
```

Streaming Chat Completion Interface – The streaming interface supports real-time interaction by delivering responses incrementally as they are generated. This enables agents to reason, act, and respond incrementally, providing immediate feedback to users and allowing downstream systems to begin processing results before the entire response is complete. The streaming mechanism reduces perceived latency and enhances the interactive experience for conversational use cases. This can be done by simply adding `stream=True` to the `client.agent.completions.create` call above.

OpenAI Compatibility – Orion implements a fully OpenAI-compatible chat completions API, ensuring seamless integration with existing tools, libraries, and workflows built around the OpenAI ecosystem. The API includes streaming support for real-time interaction, enabling low-latency responses and progressive result delivery.

```

1 import openai
2 from vlmrun.common.image import encode_image
3
4 # Initialize the OpenAI client
5 client = openai.OpenAI(base_url="https://agent.vlm.run/v1/openai", api_key="...")
6
7 # Example: Chat completion with an image input
8 messages = [
9     {"role": "user", "content": [
10        {"type": "text", "text": "What's in this image?"},
11        {"type": "image_url", "image_url": {"url": encode_image(image), "detail": "auto"}},
12    ]}
13 ]
14
15 # Perform chat completion
16 chat_completion = client.chat.completions.create(
17     model="vlmrun-orion-1:auto",
18     messages=messages,
19 )
20 print(chat_completion.choices[0].message.content)

```

Structured Outputs API – Orion provides structured output capabilities with type-safe output validation. Results conform to predefined schemas and are validated against expected types and constraints before delivery. This ensures that downstream systems receive well-formed, predictable data structures suitable for programmatic processing, reducing integration complexity and improving reliability in automated workflows. See section 3.4 for more details.

```

1 from vlmrun.client import VLRun
2 from vlmrun.common.image import encode_image
3
4 # Initialize the client
5 client = VLRun(base_url="https://agent.vlm.run/v1", api_key="...")
6
7 # Example: Chat completion with an image input
8 messages = [
9     {"role": "user", "content": [
10        {"type": "text", "text": "What's in this image?"},
11        {"type": "image_url", "image_url": {"url": encode_image(image), "detail": "auto"}},
12    ]}
13 ]
14
15 # Example response format
16 class ImageCaption(BaseModel):
17     caption: str = Field(..., description="Brief caption of the image")
18     tags: list[str] = Field(..., description="List of relevant tags (ad, news, sports)")
19
20
21 # Perform chat completion with JSON schema
22 response_format = {"type": "json_schema", "schema": ImageCaption.model_json_schema()}
23 chat_completion = client.agent.completions.create(
24     model="vlmrun-orion-1:auto",
25     messages=messages,
26     response_format=response_format,
27 )
28 content: str = chat_completion.choices[0].message.content
29 caption: ImageCaption = ImageCaption.model_validate_json(content)

```

2.4. Extensions and Future Work

Orion is designed with extensibility in mind, and several planned enhancements will further expand its capabilities and flexibility.

Tool-Calling Support – Future versions will provide the ability to call external or user-defined tools and services, enabling seamless integration of custom processing modules into the execution plan. This capability allows users to extend Orion with domain-specific algorithms, proprietary APIs, or specialized processing pipelines without modifying core system components.

Code-Generation and Execution – Planned support for automatic generation of computer-vision tools

on-the-fly will enable dynamic creation and execution of custom processing modules within the execution plan. This capability allows **Orion** to synthesize specialized tools tailored to specific task requirements, reducing the need for pre-built modules and enabling more flexible problem-solving approaches.

Deeper LLM Integration – The architecture will evolve to support a hot-swappable model architecture compatible with all leading frontier model architectures. This includes open-source model families such as LLaMA (Touvron et al., 2023a,b) and Qwen (Qwen Team, 2025; Team et al., 2024), as well as frontier model families including OpenAI GPT (Achiam et al., 2023), Anthropic Claude, and Google Gemini (et al., 2025). This flexibility enables users to select models based on performance requirements, cost constraints, deployment environments, and specific task characteristics.

3. Capabilities

3.1. Image Understanding, Reasoning and Tool-Use

Orion provides a comprehensive suite of image understanding and manipulation capabilities, enabling fine-grained visual reasoning and tool-augmented execution workflows. The capabilities are organized as follows:

3.1.1. Image Captioning

Generate dense, contextually rich captions and semantic tags for images, supporting both global scene descriptions and region-specific annotations.

Q: What is in this image?



(a) Vehicle Scene Captioning

A: A classic, light turquoise Volkswagen Beetle with chrome accents is parked on a cobblestone street, set against a warm yellow stucco wall with rustic brown wooden doors and windows. Tags: car, volkswagen, beetle, street, cobblestone, wooden, doors, windows

Q: Densely caption this image in 2 sentences.



(b) Dense captioning in Urban Environments

A: A busy city street is depicted with tall, imposing buildings lining both sides under an overcast sky, as numerous yellow taxis, vans, and other vehicles occupy the multi-lane road. Pedestrians actively cross marked crosswalks and walk along the sidewalks, while American flags adorn several buildings and red traffic lights are visible throughout the dense urban environment.

Figure 3 | Image captioning examples showing dense, rich description generation for scene understanding.

3.1.2. Visual Question Answering (VQA) with Grounding

Perform open-ended visual question answering with spatial grounding, allowing users to query arbitrary visual content and receive precise, contextually grounded responses. Orion supports both free-form natural language queries and structured interrogation 2.3 of visual elements.

Q: Can you visualize the items in the image that I need to avoid if I'm allergic to seafood?



(a) Food Allergy Visualization

A: If you are allergic to seafood, you should avoid the items highlighted in the image. I have marked the seafood items for you.

Q: Detect the light sources in the image



(b) Light Source Detection

A: I have detected and visualized all light sources in the image, identifying both natural and artificial lighting elements with their precise locations and intensities.

Figure 4 | Visual question answering examples demonstrating open-ended queries with spatial grounding and contextual responses.

3.1.3. Object, Person, and Face Detection

Perform comprehensive detection capabilities spanning multiple object categories, including general objects, faces, persons, logos, and landmarks. Provides bounding box localization, confidence scores, and class labels for downstream reasoning and analysis. Detection bounding boxes are returned in normalized xywh format.

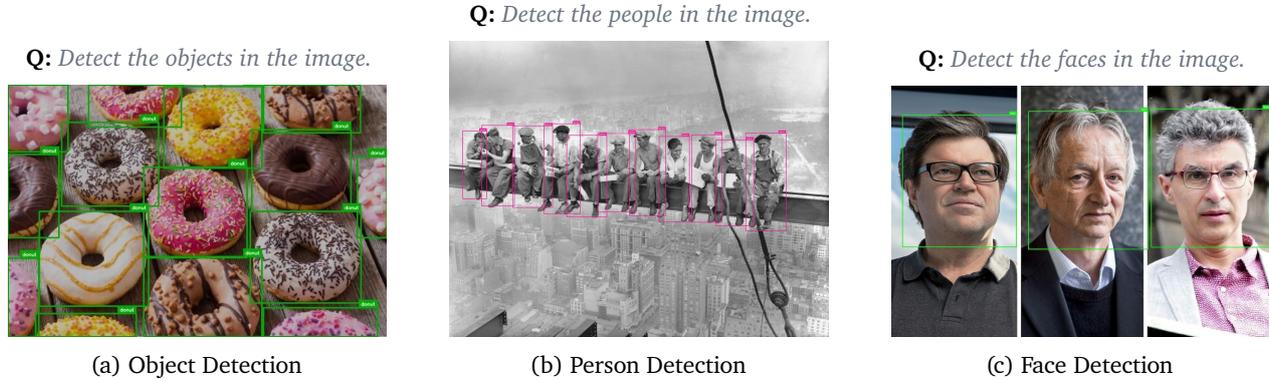


Figure 5 | Examples of object, person, and face detection capabilities showing bounding box localization with confidence scores.

3.1.4. Image Segmentation

Supports both semantic segmentation (pixel-level classification of scene elements) and instance segmentation (individual object delineation), enabling precise spatial understanding and region-based analysis for complex visual scenes. Segmentation masks are returned as uint8 images of the same size as the input image, with unique labels (0, 1, 2, ..) for each segment instance.

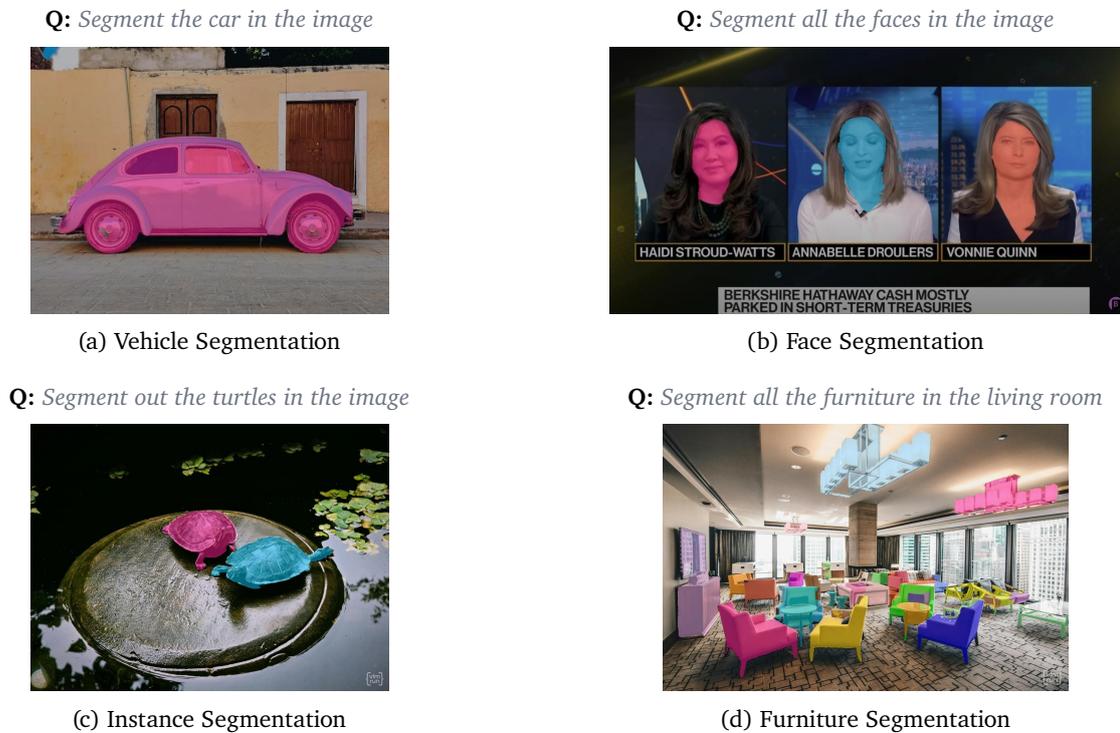


Figure 6 | Semantic and instance segmentation examples with pixel-level masks.

3.1.5. Spatial Localization and Pointing

Performs keypoint detection, classification, and 2D localization tasks, including object counting, saliency detection, and fine-grained spatial reasoning. **Orion** enables precise identification of visual anchors and regions of interest within images. Keypoint detections are returned in normalized xy format.

Q: Point to all the donuts in the image and localize them.



(a) Object Keypoint Detection

Q: Point to all the people in the image and localize them.



(b) Person Keypoint Detection

Q: Point to all the faces in the image and localize them.



(c) Face Keypoint Detection

Q: Point to all the eyes in the image.



(d) Eye Keypoint Detection

Figure 7 | Keypoint detection and localization across different object categories.

3.1.6. Image Generation and Editing

Generate, edit, remix and transform images leveraging image-generation tools. We currently support the following capabilities under **Orion**, with more coming soon:

- **Text-to-Image:** Generate high-quality images directly from natural language prompts, supporting diverse artistic styles and semantic specifications.
- **Image-to-Image:** Edit existing images based on reference images and textual instructions, enabling controlled visual transformations and content-aware modifications.
- **Image Inpainting:** Reconstruct or replace specified regions within images using contextual cues and user-provided prompts, supporting seamless content removal and insertion.
- **Style Transfer:** Transfer the artistic or visual style of one image onto another, preserving content structure while adapting stylistic attributes.

Notable, each of the above image generation capabilities are not restricted to single image inputs, but can support multiple inputs or inputs derived from outputs of other tools allowing for true compositionality.

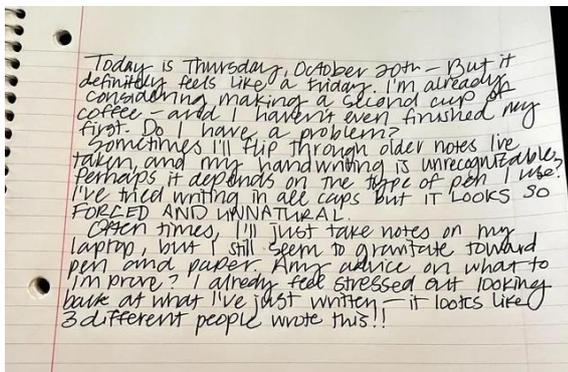


Figure 8 | Image generation and editing capabilities including text-to-image synthesis, image transformation, inpainting, and style transfer.

3.1.7. Image OCR

Robust in-the-wild optical character recognition (OCR) capable of extracting text, bounding boxes and word-level confidence scores from challenging visual contexts, including license plates, street signs, logos, product labels, and other unstructured text sources. **Orion** handles diverse fonts, orientations, and lighting conditions.

Q: Read the text in the image.



(a) Handwritten Text Extraction

A: Today is Thursday, October 20th – But it definitely feels like a Friday. I’m already considering making a second cup of coffee – and I haven’t even finished my first. Do I have a problem? Sometimes I’ll flip through older notes I’ve taken, and my handwriting is unrecognizable. Perhaps it depends on the type of pen I use. I’ve tried writing in all caps but IT LOOKS SO FORCED AND UNNATURAL. Often times, I’ll just take notes on my laptop, but I still seem to gravitate toward pen and paper. Any advice on what to improve? I already feel stressed out looking back at what I’ve just written – it looks like 3 different people wrote this!!

Q: Read the text in the image.



(b) News Article Text Extraction

A: All the News That’s Fit to Print The New York Times LATE CITY EDITION Werdert Fait, want today’y aleeer 0-30 35-46. Tengo-Hum, Lodre yesterday 4A Complete V.I. rport en 7. 30. VOL.CXVIII.No.40,321 NEW YORK, MONDAY, JULY 21, 1959 10 CENTS MEN WALK ON MOON ASTRONAUTS LAND ON PLAIN; COLLECT ROCKS, PLANT FLAG. Voice From Moon: "Eagle Has Landed" ... (truncated for brevity).

Figure 9 | In-the-wild OCR examples showing robust text extraction from challenging visual contexts with diverse fonts and layouts.

3.1.8. UI Parsing

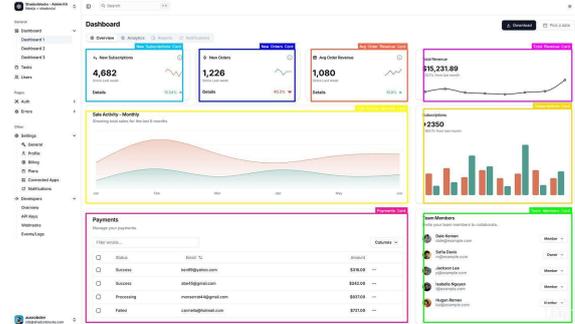
Automatically parse and extract user interface elements from screenshots or application images, including buttons, text fields, links, icons, and other interactive components. **Orion** supports accessibility workflows, automated testing, and interface analysis. UI bounding boxes are returned in normalized `xywh` format.

Q: Detect and visualize all the UI elements in this screenshot.



(a) UI Element Detection and Parsing

Q: Detect all the main UI elements excluding the sidebar and visualize the boxes.



(b) Card-based Interface

Figure 10 | UI parsing examples showing automatic extraction of interface elements including buttons, text fields, and interactive components.

3.1.9. Image Tools

Extend and compose visual workflows with pre-processing tools such as cropping, rotation, blurring, super-resolution, and other transformations. Users will soon be able to integrate custom image processing modules to extend the system’s analytical and generative capabilities.



(a) Original Image



(b) Cropped Image



(c) Rotated Image



(d) Crop + VQA

Q: (a) Crop the clock in this image (b) Rotate the image 90 degrees counterclockwise.

A: (b) I have cropped to the specific region. (c) I have rotated the image by 90 degrees counter-clockwise.

Q: (a) Crop into the clock in the image and extract the time shown.

A: (d) I have cropped the clock region from the image and extracted the time shown, providing both the cropped image and the extracted time value.

Figure 11 | Image transformation tools including cropping, rotation, enhancement, and time extraction.

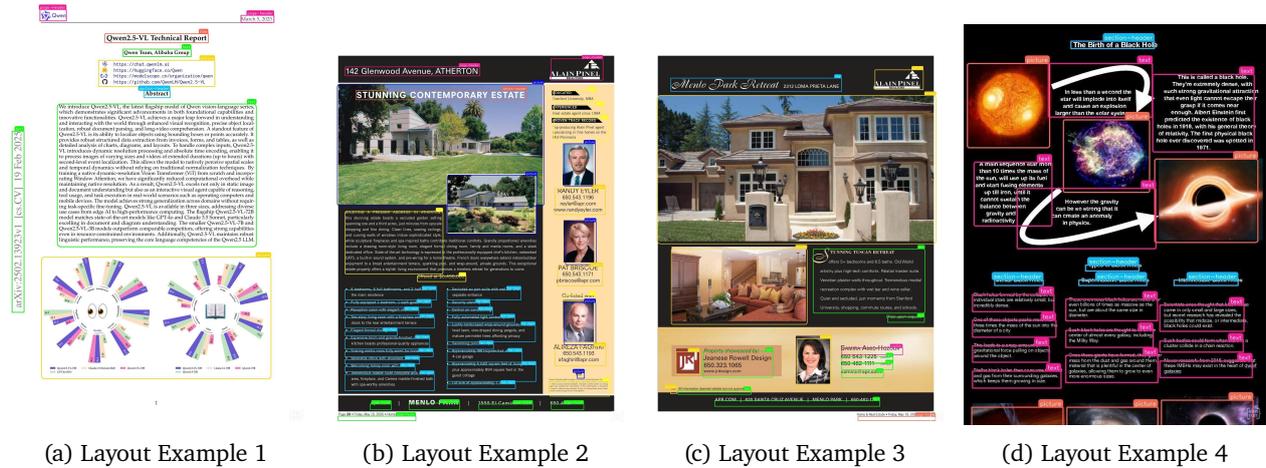
These capabilities enable **Orion** to serve as a versatile visual intelligence platform, supporting wide range of visual AI applications.

3.2. Document Understanding, Reasoning and Tool-Use

Orion provides extensive document understanding capabilities, supporting a variety of complex structure-aware tasks and extractive workflows. The system is designed to handle both scanned and digital documents, enabling robust analysis across diverse document types and formats. Our document understanding capabilities can be organized into the following functional categories:

3.2.1. Document Layout Understanding

Robustly analyze complex document layouts, extracting structural hierarchies including paragraphs, headers, tables, footnotes, and other semantic elements from both scanned and digital sources. The system preserves reading order and spatial relationships to maintain document coherence. Detection bounding boxes are returned in normalized xywh format.



- Q:** Analyze the document layout and extract its structural elements.
A: I have analyzed the document layout and extracted the structural hierarchy, identifying headers, paragraphs, tables, footnotes, and other semantic elements while preserving the reading order and spatial relationships.
- Q:** Get the layout of the document and visualize all the text.
A: I have extracted the complete document layout and visualized all text elements with their precise locations, showing the spatial organization of content including text blocks, titles, and formatting structures.

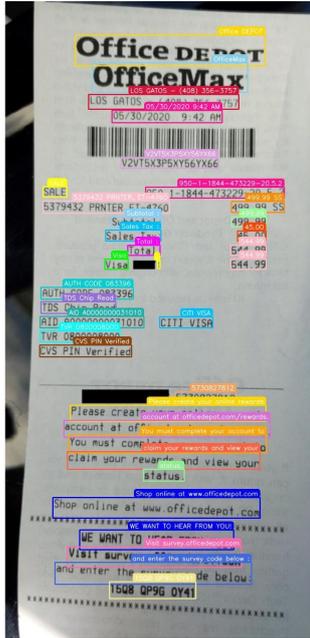
Figure 12 | Document layout analysis showing detection of structural elements including headers, paragraphs, tables, and figures with preserved reading order.

3.2.2. Document OCR

Perform high-accuracy optical character recognition with specialized support for structured forms, tables, and handwritten text. The system handles both printed and handwritten content, enabling robust text extraction across diverse document types and quality levels.

- **Form OCR:** Localize form fields, tables, checkboxes, signatures, and visual anchors in documents—enabling workflows such as form field extraction and field-value alignment, including multi-page analysis and cross-page field tracking.

Q: Localize form fields and align them with their values.



(a) Form Extraction

Q: Parse this document and extract all the handwritten fields. Detect the locations of each of the fields and visualize them.

Physician Referral Form

Referring Physician Details			
First Name <i>Krene</i>	Last Name <i>Wong</i>	Society <i>family health</i>	
Email <i>krenewy@gmail.com</i>		Preferred Phone Number <i>6753367412</i>	
Patient Details			
First Name <i>Kevin</i>	Last Name <i>Chan</i>	Date of Birth <i>9/15/2001</i>	
Email <i>kevinchan@gmail.com</i>		Preferred Phone Number <i>6153012311</i>	
Diagnosis <i>back pain</i>			
Referral Reason <i>need further diagnose</i>			
Details about the patient's condition <i>sustained back pain that doesn't go away after initial treatment</i>			
Why does the patient need to be seen by another physician? <i>more diagnose</i>			
Referred Physician Details			
First Name <i>Sam</i>	Last Name <i>Yong</i>	Society <i>family health</i>	
Email <i>Sam@gmail.com</i>		Preferred Phone Number <i>651637234</i>	

http://carepatron.com Powered by carepatron

(b) Handwritten Form Extraction

Figure 13 | Visual grounding examples showing form field localization and field-value alignment with bounding box coordinates.

3.2.3. Document Tools

Provide specialized modules for large-scale document processing, enabling efficient manipulation and analysis of extensive document collections. This includes intelligent pagination, in-place editing, and content redaction capabilities.

- **Pagination:** Intelligently index and retrieve only relevant pages of a document, enabling efficient agentic processing of large documents (100+ pages) without loading entire files into memory.
- **Editing / Redaction:** Edit or redact the document in-place, enabling the agent to modify the document as it is being processed.

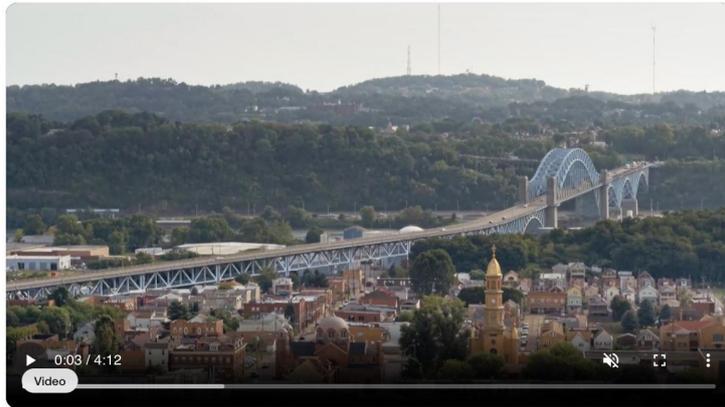
These capabilities enable **orion** to serve as a comprehensive document intelligence platform, supporting use cases ranging from automated form processing and compliance workflows to large-scale document review and knowledge extraction.

3.3. Video Understanding, Reasoning and Tool-Use

Orion provides comprehensive capabilities for video analysis, generation, and tool-augmented workflows. The system supports a diverse range of video modalities and tasks, organized into the following functional categories:

3.3.1. Video Captioning

Automatically generate detailed descriptions and scene-level captions for entire videos and video segments. Supports multimodal prompts, per-frame analysis, and temporal alignment of captions with video timestamps, enabling fine-grained narrative generation.



(a) Video Input [[link](#)]

Q: Parse the video and return a markdown summary with key scenes and timestamps.

```
Here is a markdown summary of the key scenes and timestamps:
00:00 - 00:06: Establishing shot of a city with a large blue bridge and a town in the
↳ foreground.
00:06 - 00:13: Scott Baker is interviewed in a modern bakery setting, discussing his
↳ family's long-standing connection to the community.
00:10 - 00:15: Archival black and white footage shows a bustling bakery counter with
↳ cakes and pastries.
00:14 - 00:16: A vintage black and white photo of a Jenny Lee Baked Goods delivery truck
↳ with two men.
00:16 - 00:19: Exterior view of a "Jenny Lee Bakery" storefront.
00:19 - 00:31: Donna is interviewed in a bakery setting, sharing her experiences.
00:32 - 00:34: A vintage black and white photo of three men wearing chef hats.
00:35 - 00:44: Bernie is interviewed in a bakery setting, reflecting on his career.
00:39 - 00:44: A vintage black and white photo of children dressed in chef attire.
00:45 - 00:48: Two men are seen reviewing documents in front of shelves stocked with
↳ bread.
00:49 - 00:58: A train passes by under a large blue arched bridge.
01:01 - 01:10: Donna recounts a difficult period for the business.
... (truncated for brevity).
```

Figure 14 | Video captioning with temporal grounding. See [full chat history](#).

3.3.2. Temporal Grounding

Localize and identify key moments or events within a video based on natural language instructions, providing precise start and end timestamps for any described action, object, or scene. This capability is particularly useful for tasks such as highlight extraction and video summarization. Temporal grounding is provided in the HH:MM:SS format and can also be used for sampling and trimming clips using the tools mentioned below.

Q: Get the best highlight showing the most fireworks in this video.



(a) Fireworks highlight extraction

A: I have identified the most spectacular highlight of the fireworks display, featuring a breathtaking finale where the entire skyline is engulfed in a vibrant red glow, with precise timestamps (00:23 to 00:28).

Q: Step 1: Detect and highlight all people present in the image. Step 2: Apply deoldification techniques to restore true-to-life color and detail. Return a high-quality, artifact-free video clearly demonstrating the deoldified scene, focusing on preserving facial features and overall realism.



(b) Video deoldification

A: I have detected and highlighted all people in the image, then applied deoldification techniques to restore true-to-life color and detail, producing a high-quality video demonstrating the enhanced scene with preserved facial features and overall realism.

Figure 15 | Temporal grounding examples showing highlight extraction and video enhancement.

3.3.3. Video Generation

Create short video sequences from text prompts or reference images, supporting open-domain video synthesis, style transfer, and dynamic scene animation.

3.3.4. Video Tools

Extensible toolkit for video editing and analysis, including frame sampling, highlight extraction, time-based trimming, and scene segmentation. Users can integrate custom video processing modules to extend the system’s analytical and generative capabilities.

- *Frame Sampling:* Extract frames at regular intervals or specific timestamps for tasks such as thumbnail generation and visual analysis. See Figure 16.
- *Highlight Extraction:* Automatically identify and extract the most salient moments or key actions in a video, returning their start and end times along with segment URLs. See Figure 16.
- *Time-Based Trimming:* Precisely trims videos to specified time segments, supporting millisecond-level accuracy.



(a) Frame 1



(b) Frame 2



(c) Frame 3



(d) Frame 4

Q: Analyze the video content and extract key frames for captioning.

A: I have sampled key frames from the video and can generate detailed descriptions for each frame, enabling temporal analysis and scene understanding. The frame sampling provides a comprehensive view of the video content for captioning and analysis.

Figure 16 | Video frame sampling examples showing temporal analysis and keyframe extraction for video captioning and scene understanding.

All video workflows can be composed into multi-step pipelines, enabling rich, agentic parsing, annotation, and content creation for diverse use-cases across domains such as media, healthcare, and enterprise document review.

3.4. Structured Outputs

A core capability of **Orion** is robust, type-safe structured outputs for every step of agentic reasoning and execution. Unlike conventional LLM systems that produce free-form text, **Orion** emits outputs in strict, schema-validated formats—enabling precise downstream processing, reliable tool chaining, and seamless integration into applications and APIs.

- **Schema-driven Outputs** – Agents and tools are driven by explicit schemas, defining the required output types (e.g., arrays, key-value objects, tagged regions, time segments, entities). This ensures each step can be programmatically consumed and reliably orchestrated.
- **Multi-modal Structured Data** – Multimedia outputs, returned in the form of pre-signed URLs, can include mixed-modality values (e.g., image crops, video clips, or generated documents) in a single response, enabling unified downstream processing and interleaving of modalities across agent workflows.
- **API Ready and OpenAI-Compatible** – All outputs conform to Pydantic-compatible JSON output formats, supporting direct integration into developer pipelines, evaluation systems, or end-user applications with minimal custom handling.

This structured output paradigm unlocks agentic planning, iterative tool invocation, reliable evaluation, and advanced workflow composition within **Orion**—transforming visual reasoning from opaque black-box outputs to transparent and verifiable computation.

4. Quantitative Evaluation

We evaluate **Orion** across image, document, and video inputs, comparing its understanding, reasoning, and execution capabilities, against all frontier vision-language models. The benchmarks measure performance on diverse tasks including (but not restricted to) visual question answering, accurate text/object/face detection in images, document/video parsing and summarization, and more.

4.1. Benchmark Performance

Our evaluation spans established multi-modal benchmarks including MMMU (Yue et al., 2024), MMBench (Liu et al., 2024), DocVQA (Mathew et al., 2021), and other comprehensive evaluation suites (Liu et al., 2023). In the coming weeks, we will be updating these sections with additional benchmarks and updated numbers as our agent evolves.

Capability	Benchmark	Orion	Gemini 2.5 Flash	GPT-5 Mini	Claude Opus-4.1
STEM & Puzzle	MMMU _{val}	72.9	72.7	67.9	–
	MMMU_Pro	58.2	60.7	53.7	60.7
General VQA	MMBench EN (dev)	86.3	82.4	78.5	84.1
	RealWorldQA	69.2	70.6	73.3	68.5
	MMStar	68.4	71.3	61.3	71.0
Hallucination	HallusionBench	69.7	53.6	55.9	55.1
	MM_MT_Bench	8.2	7.1	7.4	7.9
Docs & Charts	AI2D _{TEST}	85.2	84.8	82.9	84.4
	MMLongBench-Doc	33.3	38.3	42.4	48.1
	OCRBench	798	813	807	750
Multi-image Reasoning	BLINK	64.2	62.0	56.7	62.9
	MUIREBENCH	67.5	67.0	57.5	–

Table 2 | Comparison of **Orion** against leading VLMs on selected benchmarks. Bold indicates highest score per benchmark.

4.2. Human Evaluation Methodology

To comprehensively assess **Orion**’s performance across diverse visual tasks, we conducted a qualitative human evaluation comparing outputs from **Orion**, ChatGPT (GPT-5), Gemini 2.5 Pro, and Claude Sonnet 4.5 across 46 diverse visual tasks. This evaluation methodology was designed to provide rigorous, unbiased assessment of model outputs across a wide spectrum of visual AI capabilities, from object detection and segmentation to complex multi-step visual reasoning, document analysis, and image generation.

Evaluation Design. To ensure objective assessment, we implement a double-blind evaluation protocol where all model outputs are presented to evaluators in randomized order, labeled only as Model A, B, C, or D. Each task is evaluated by at least three independent evaluators who undergo calibration training to establish consistent scoring standards.

Composite Overall Score. We develop a composite overall score (0–10) that aggregates multiple dimensions of output quality. The score is calculated as a weighted combination of four components:

- *Helpfulness* (30%), measuring whether the model addresses all the requested requirements;
- *Correctness* (35%), assessing factual correctness;
- *Presentation* (20%, for visual outputs), evaluating presentation and artifact-free rendering; and

- *Instruction Following* (15%), determining whether the output meets specified constraints and requirements.

Evaluation Process. Ten independent evaluators assess 46 diverse visual tasks spanning object detection, multi-step reasoning, medical image analysis, document processing, and image generation. For each task, evaluators receive the input and prompt, along with outputs from all four models in randomized order, and score each output independently across all four dimensions.

Results and Analysis. The human evaluation results, visualized in Figure 17, demonstrate **Orion**'s superior performance across the majority of evaluated tasks. The double-blind design and multi-dimensional scoring ensure these results reflect genuine performance differences, establishing **Orion** as the leading system for production-grade visual tasks requiring precision, multi-step reasoning, and specialized tool integration.

4.3. Performance Analysis and Comparison

As shown in Table 2, **Orion** demonstrates superior performance across multiple vision capabilities when compared to frontier vision-language models. The comprehensive benchmark results reveal that **Orion**'s agentic architecture, which orchestrates specialized computer vision tools, enables it to achieve more accurate and reliable performance across diverse visual understanding tasks compared to monolithic models. For an interactive version of the results showcased in the table 2, see our showdown page at <https://chat.vlm.run/showdown>.

Reduced Hallucination and Enhanced Accuracy **Orion** exhibits significantly lower hallucination rates compared to other frontier models, as evidenced by its state-of-the-art performance on hallucination detection benchmarks. The system's ability to leverage precise visual analysis tools—including object detection, segmentation, and spatial reasoning—enables it to ground its responses in actual visual content rather than generating hallucinated details. This tool-augmented approach provides built-in verification mechanisms that help the system distinguish between real visual information and spurious patterns, resulting in more trustworthy and accurate outputs across all vision tasks.

Superior Multi-Capability Vision Performance. **Orion** demonstrates enhanced accuracy across multiple vision capabilities compared to other frontier models. The system's specialized tool suite enables precise execution of diverse visual tasks including object detection and localization, semantic and instance segmentation, optical character recognition, spatial reasoning, and multi-image understanding. Unlike monolithic models that rely solely on learned representations, **Orion** can invoke specialized tools to perform exact visual computations—such as precise bounding box detection, pixel-level segmentation masks, and character-level OCR—resulting in more accurate results across these fundamental vision capabilities.

Comprehensive Visual Understanding. The benchmark results demonstrate that **Orion** achieves competitive or superior performance across a wide spectrum of visual understanding tasks, from general visual question answering to specialized document analysis, chart interpretation, and multi-image reasoning. The system's ability to seamlessly integrate multiple vision tools within a single agentic workflow enables it to handle complex visual scenarios that require combining multiple capabilities—such as detecting objects, extracting text, analyzing spatial relationships, and reasoning about visual content—more accurately than models that attempt to perform all these tasks through a single neural architecture.

Model Performance Comparison Across Evaluation Metrics



Figure 17 | Comprehensive evaluation results showing **Orion**'s superior performance across 46 diverse visual tasks compared to frontier Vision-Language Models. The evaluation demonstrates **Orion**'s consistent advantage in complex visual reasoning, specialized skill execution, and multi-step workflow completion.

Production-Ready Reliability. The consistent performance improvements demonstrated across benchmarks indicate that **Orion**'s tool-augmented approach provides more reliable and accurate vision capabilities suitable for production deployment. The system's reduced hallucination rates and enhanced accuracy across multiple vision tasks make it particularly well-suited for real-world applications where precision and trustworthiness are critical. By combining neural reasoning with symbolic visual tools, **Orion** achieves a balance between the flexibility of learned models and the precision of specialized computer vision algorithms, resulting in superior overall performance on image understanding and analysis tasks.

4.4. Future Work

While **Orion** demonstrates significant capabilities in multi-modal reasoning and tool use, several directions remain for future exploration:

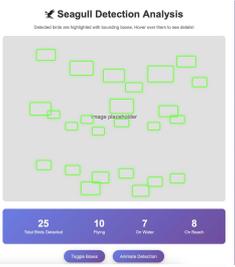
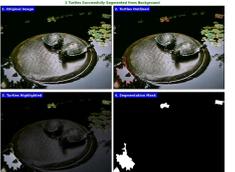
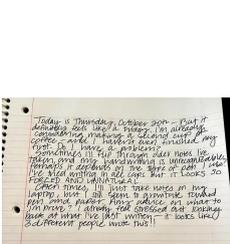
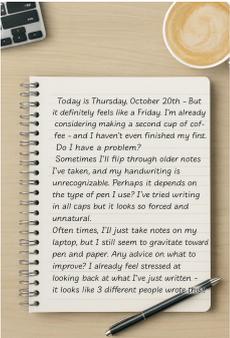
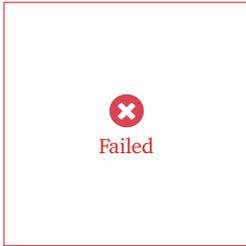
Enhanced Benchmarking and Evaluation. Current multi-modal benchmarks predominantly focus on single-turn interactions and relatively simple tasks. This limitation stems largely from the difficulty of evaluating complex, multi-step outputs that require visual verification. We are developing a comprehensive suite of benchmarks specifically designed to assess performance on intricate, multi-turn tasks where outputs must be visually validated against ground truth.

Improved Tool Integration. Expanding the repertoire of available tools and refining the tool selection mechanism could further enhance the system's capabilities. This includes integrating domain-specific tools for specialized applications and developing more sophisticated strategies for tool composition.

Efficiency Optimizations. While the current system achieves strong performance, there remain opportunities to optimize computational efficiency, particularly in reducing the number of inference calls required for complex tasks and improving the speed of tool execution.

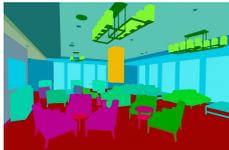
Robustness and Error Recovery. Enhancing the system's ability to detect and recover from errors, particularly in long-running multi-step tasks, represents an important area for improvement. This includes developing better mechanisms for self-correction and validation of intermediate outputs.

Table 3 | Comparison of **Orion** with frontier vision-language models across diverse tasks. See our interactive showdown at <https://chat.vlm.run/showdown>.

Input	Orion	GPT 5	Gemini 2.5	Claude 4.5
				
	1. Detect all the birds in the image and visualize them.			
				
	2. Segment out the turtles in the image.			
				
	3. Identify and list all major rooms in the floor plan, providing the square footage for each. Redesign the space using Japandi-style furnishings, emphasizing natural light, neutral color palettes, and minimalist forms. Return an updated floor plan with labeled rooms and suggested furniture layout.			
				
	4. Enhance and beautify the handwritten text in the document, so it appears clear and aesthetically pleasing. Replace the entire background with a realistic desk scene that includes a laptop and a coffee cup. Make the text look as if it is neatly written in a notebook, with a pen and the coffee cup nearby. Ensure the final image is high-quality, artifact-free, and visually appealing.			
				
	5. Detect and extract only the most important details in the news article title, such as key subjects, events, or names, and present them in a markdown format. Only show the title and dates from the image overlaid with bounding boxes.			

Continued on next page...

Table 3 | Comparison of **Orion** with frontier vision-language models across diverse tasks. See our interactive showdown at <https://chat.vlm.run/showdown>.(continued)

Input	Orion	GPT 5	Gemini 2.5	Claude 4.5
				
11. Generate a new creative image inspired by this reference, but with a completely different artistic style and composition.				
				
12. Segment all the furniture in the living room.				
				
13. Detect all the cars and pedestrians at the intersection.				
				
14. Detect the light sources in the image.				
				
15. Detect all the UI elements in this image and visualize the boxes.				
				
16. Segment out the faces in this image and visualize the segments.				

Continued on next page...

Table 3 | Comparison of **Orion** with frontier vision-language models across diverse tasks. See our interactive showdown at <https://chat.vlm.run/showdown>.(continued)

Input	Orion	GPT 5	Gemini 2.5	Claude 4.5
<p>17. Generate an image of this girl on the mars. I want her in front of a futuristic settlement house on mars as if she has been there for her whole lif enjoying the day. I want her outfit and pose to blend into the surrounding and look natural.</p>				
<p>18. Give a detailed explanation of the plan and generate detailed 3D architectural visualizations and interior design concepts based on the plan.</p>				
<p>19. Generate detailed 3D architectural visualizations and interior design concepts based on this floor plan, including furniture placement and lighting.</p>				
			<div style="border: 1px solid red; padding: 20px; text-align: center;"> ✘ Failed </div>	<div style="border: 1px solid red; padding: 20px; text-align: center;"> ✘ Failed </div>
<p>20. Get the layout of the document and visualize all the text.</p>				

Continued on next page...

Table 3 | Comparison of **Orion** with frontier vision-language models across diverse tasks. See our interactive showdown at <https://chat.vlm.run/showdown>.(continued)

Input	Orion	GPT 5	Gemini 2.5	Claude 4.5

21. Can you blur out the patient name, dob, sex and marriage status?

Input	Orion	GPT 5	Gemini 2.5	Claude 4.5

22. Parse this document and extract all the handwritten fields. Detect the locations of each of the fields and visualize them.



23. Crop into the clock in the image and extract the time shown.



24. First detect the license plates in the image and blur them.

Continued on next page...

Table 3 | Comparison of **Orion** with frontier vision-language models across diverse tasks. See our interactive showdown at <https://chat.vlm.run/showdown>.(continued)

Input	Orion	GPT 5	Gemini 2.5	Claude 4.5
25. Analyze this image quality and enhance the image to a high resolution.				
26. Extract the first and last page from the document, visualize and analyze them.				
27. Get the best highlight showing the most fireworks in this video.				
28. Extract any text visible in the document, their locations and visualize them.				

5. Discussion and Future Work

Orion represents a new paradigm in visual AI: **agentic tool-augmented reasoning** that combines the flexibility of large vision-language models (Li et al., 2023a,b; Radford et al., 2021) with the precision of specialized computer vision tools, extending the principles of tool-augmented language models (Patil et al., 2023; Schick et al., 2023) to the visual domain.

5.1. Limitations and Challenges

While **Orion** demonstrates strong performance, several challenges remain:

Tool Selection Accuracy – In complex scenarios with overlapping tool capabilities, the agent occasionally selects suboptimal tools. Future work will focus on learning from historical tool selection patterns, implementing cost-aware tool selection, and enabling dynamic tool specialization based on domain-specific requirements.

Long-Horizon Planning – For workflows exceeding 30+ steps, success rates decline due to error accumulation across many steps, difficulty maintaining coherent state over long sequences, and limited ability to backtrack and revise early decisions.

Computational Cost – While referencing multi-modal objects improve efficiency, complex workflows still require significant compute. Multiple LLM calls for planning and reflection, parallel tool execution infrastructure requirements, and the trade-off between quality and cost remain areas for optimization.

5.2. Broader Impact

The introduction of tool-augmented visual agents has implications beyond technical performance.

Democratizing Computer Vision – **Orion** makes sophisticated computer-vision capabilities accessible to non-experts, enabling rapid prototyping of visual applications, allowing domain experts to build custom workflows without engineering teams, and reducing the barrier to entry for visual AI adoption.

Interpretability and Trust – Unlike black-box models, **Orion** provides transparent execution traces showing tool usage, explicit reasoning about tool selection, and the ability to audit and verify each step, fostering greater trust and accountability.

Ethical Considerations – The system’s capabilities require careful consideration of privacy implications related to face recognition and tracking, potential misuse for surveillance or manipulation, bias propagation through tool selection and execution, and the need for human oversight in sensitive applications. We are committed to responsible development and deployment, working with domain experts and ethicists to establish appropriate safeguards.

Contributors

The development of **Orion** was performed by the VLM Run Research¹ team – Dinesh Narapureddy and Sudeep Pillai. We gratefully acknowledge the following team members for their contributions to the development of **Orion**: Shahrear Bin Amin, Mirajul Mohin, and Dylan Snyder for the development of the frontend chat interface, including backend system development; and Lona Kiragu for contributing to dataset evaluation and scoring. Their collective efforts were essential to the development and deployment of **Orion**.

References

J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

¹research@vlm.run

- C. et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL <https://arxiv.org/abs/2507.06261>.
- S. Lee, S. Kim, S. H. Park, G. Kim, and M. Seo. Prometheus-vision: Vision-language model as a judge for fine-grained evaluation, 2024. URL <https://arxiv.org/abs/2401.06591>.
- C. Li, L. Li, J. Yan, X. Weng, K. Chu, J. Chen, L. Yu, Y. Chang, H. Xu, H. Zhang, et al. Llava: Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023a.
- J. Li, D. Li, S. Savarese, and S. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023b.
- H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2023.
- Y. Liu, Z. L. Li, H. Li, H. Yu, F. Yuan, C. Jiang, Z. Yan, J. Shen, L. Sun, J. Shao, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2024.
- M. Mathew, D. Karatzas, and C. Jawahar. Docvqa: A dataset for vqa on document images. In *WACV*, 2021.
- S. G. Patil, T. Zhang, X. Wang, and J. E. Gonzalez. Gorilla: Large language model connected with massive apis. *arXiv preprint arXiv:2305.15334*, 2023.
- Qwen Team. Qwen3 technical report, 2025. URL <https://arxiv.org/abs/2505.09388>.
- A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, and T. Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36, 2023.
- Q. Team et al. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12179*, 2024.
- H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.
- H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.
- J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837, 2022.
- S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2023.
- X. Yue, Y. Ni, K. Zhang, T. Zheng, R. Liu, G. Zhang, S. Stevens, D. Jiang, W. Ren, Y. Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024.