MCP SERVER | NO CODE | CLOUD HOSTED

EyePop.ai MCP

Turn video feeds into structured data instantly.

EyePop.ai lets your agent run computer vision tasks on images and video streams. It provides pre-trained models for detecting objects, recognizing faces, and classifying visual content in real time. You can analyze media feeds and manage complex visual pipelines through natural language commands.

A+ Quality Score 100/100

computer-vision | object-detection | face-recognition | visual-intelligence | media-streams | api-integration

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/eyepopai

SCAN TO VIEW

The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect to the software your business already runs. We handle the hosting, the security, the credentials, the uptime — you get agents that actually do things.

We operate the world's largest managed MCP catalog. Major SaaS platforms, CRMs, databases, and cloud providers — running, monitored, production-ready. This MCP server is hosted and maintained by the Vinkius Cloud for AI Agents.

> *The agent doesn't manage credentials, doesn't manage uptime, doesn't manage security. Vinkius does.*
>
> — Architecture principle

Four Pillars of the Vinkius Runtime

01 — Security by design

Credentials stay encrypted at rest via AES-256. The AI agent never touches raw keys — they're injected into a sandboxed V8 isolate at runtime. Actions are logged, and connections have an emergency kill switch.

02 — Built on MCP Fusion

This MCP server was built with **MCP Fusion**, the open-source framework (Apache 2.0) that powers the entire Vinkius catalog. Schema-as-firewall strips undeclared fields, compiled PII redaction runs at zero overhead, and cryptographic lockfiles produce git-diffable audit trails.

03 — Deterministic observability

Eight immutable metrics per endpoint: request volume, p95 latency, error rate, active connections, cost attribution. A live payload feed logs every tool call with mutation detection.

04 — Autonomous operations

Servers are deployed, monitored, and patched autonomously. New capabilities and security patches ship weekly. Zero-downtime deployments ensure continuous availability across all managed MCP servers.

AES-256	Ed25519	24h TTL	V8 Isolate
Encryption at rest	PKI vault signatures	Ephemeral session keys	Sandboxed execution

One Token. Instant Access.

Every MCP server on Vinkius is accessed through a **Connection Token**. Tokens are generated in the cloud dashboard and produce a unique MCP endpoint URL. Paste this URL into any MCP-compatible client — no SDK required.

A single token can serve **multiple AI clients simultaneously**, or you can issue separate tokens per client for granular access control. Each token tracks its own request count, last activity timestamp, and can be individually enabled or revoked.

MCP ENDPOINT

```
https://edge.vinkius.com/{token}/mcp
```

Claude · Cursor · VS Code · Windsurf · Grok · Gemini

Security Is the Architecture

Security in Vinkius is not a feature — it's the foundation of the runtime. The gateway enforces multiple independent protection layers between AI agents and third-party APIs.

01 — Ed25519 PKI Vault

Every workspace has an Ed25519 Master Key. Session keys are generated ephemerally (24h TTL) and signed by the Master Key. Credentials never leave the vault boundary.

02 — V8 Isolate Sandboxing

Tool code runs inside isolated-vm V8 isolates with 64 MB memory caps and per-request timeouts. No filesystem access, no network access except through the SSRF-guarded fetch bridge.

03 — SSRF Guard

All outbound HTTP requests are DNS-resolved and validated before execution. Private IP ranges (10.x, 172.16-31.x, 192.168.x, AWS metadata 169.254.x) are blocked at the network layer.

04 — DLP & PII Redaction

A ResponseGuard pipeline intercepts every tool response. Configurable redaction patterns strip sensitive fields (emails, SSNs, card numbers) before data reaches the AI agent.

05 — Cryptographic Audit Trail

Every request is signed into a SHA-256 hash chain with Ed25519 signatures. Events form a tamper-proof, SIEM-exportable forensic record.

06 — Honeytoken Trap System

Phantom credentials are injected into isolated environments. If a honeytoken is used outside Vinkius infrastructure, the server is quarantined instantly.

Emergency Kill Switch

EU AI Act Art. 14(1) Compliant

The kill switch is an **emergency halt** mechanism — not a simple toggle. When triggered, it executes three actions atomically:

01 — Server deactivated

The MCP server is immediately taken offline across the entire cluster.

02 — All tokens revoked

Every connection token is invalidated. Total lockout — reconnection blocked until new tokens are issued.

03 — WebSocket connections killed

Active connections terminated via Redis pubsub broadcast. Propagates to every runtime node in the cluster.

Full Visibility. Zero Guesswork.

The Vinkius cloud dashboard includes a full MCP Governance suite — real-time analytics and security controls for production AI operations.

Control Plane

KPI dashboard with request volume, latency, success rate, token consumption, and AI-generated operational briefings.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Incident Log

Error trends, failure rates, status-code breakdowns, and forensic audit trail access.

> Get started at **cloud.vinkius.com** — connect your AI agent in under 60 seconds.

EyePop.ai MCP

10 tools available
Cloud-hosted on Vinkius

Need to understand what's happening inside a live camera feed or a batch of photos? This MCP connects your agent directly to EyePop.ai’s visual intelligence engine. Instead of manually reviewing thousands of frames, you tell your AI client to process the media and it sends back structured data. You can programmatically create, monitor, and manage entire visual processing pipelines, getting real-time updates on detected objects and their confidence scores.

It's like having a dedicated computer vision architect built into your agent workflow. Whether you need simple object detection or complex multi-stage analysis, this MCP handles the heavy lifting. Once connected via Vinkius, your agent can use natural conversation to list existing visual models, check pipeline statuses, and even retrieve specific coordinates for bounding boxes, keeping all your data perfectly coordinated without needing to jump between dashboards.

Core Capabilities

01 — Analyze static images

Runs object detection and labeling on single pictures.

02 — Track objects in video streams

Monitors and detects items across an entire sequence of recorded frames.

03 — Manage visual pipelines (Pops)

Creates, lists, and monitors the status of your complex computer vision workflows.

04 — Monitor system health

Verifies API connectivity and tracks processing volumes to ensure reliable service scaling.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/eyepopai** — connect your AI agent in three steps.

01 Subscribe to the MCP on Vinkius, then retrieve your unique EyePop.ai API Key from your dashboard.

02 Connect this MCP to any compatible agent—like Cursor or Claude—to start orchestrating visual intelligence using natural conversation.

03 Ask your agent a question like, 'List all active pipelines and check the detection metadata for the store entrance,' and it runs the commands automatically.

The bottom line is that you use conversational prompts to trigger complex, multi-step computer vision operations on visual media.

Built For

This MCP is for technical teams who deal with large volumes of video or image data. It's the ops engineer tired of clicking through dozens of dashboards to get a simple status update, and the retail analyst who needs immediate object counts without leaving their core workspace.

Security Operations Manager

Uses this MCP to instantly pull detection summaries from live feeds or historical recordings using natural language commands.

Retail Data Analyst

Runs visual analysis on foot traffic and customer movement data, retrieving bounding box coordinates to quantify behavior patterns.

Developer/Integrator

Connects the high-speed vision data into custom monitoring tools by querying specific object labels or API status checks.

What Changes When You Connect

01 Analyze media streams without manual review. Use the `analyze_video` tool to get temporal object detection results, tracking what happens across hours of footage automatically.

02 Maintain a perfect visual knowledge pipeline by using `create_pop`. This lets you set up recurring analysis jobs and monitor their status via natural language queries.

03 Get precise data points instantly. Use the MCP to retrieve bounding box coordinates and classification IDs, giving you structured records for every object found.

04 Monitor your entire system health through the agent. The `check_eyepop_status` tool verifies API connectivity so you never lose valuable processing time due to a simple outage.

05 Manage complex resources using `list_pops`. Instead of diving into dashboards, just ask your agent for an overview of all active and inactive pipelines in seconds.

Real-World Applications

Analyzing post-incident video evidence

A security manager needs to know exactly what happened at the loading dock. Instead of manually scrubbing through hours of footage, they ask their agent to run `analyze_video` on the relevant clip. The agent returns a structured list of object detections, labeling vehicles and personnel with confidence scores.

Debugging a visual processing system

A developer wants to check if their new detection model is working correctly. They first use `list_models` to verify the correct version, then use `get_model` to retrieve its specifications before running an integration test.

Tracking customer flow in retail spaces

A retail analyst wants to know if people are spending enough time near certain displays. They use the MCP to run `analyze_image` on static camera snapshots, getting precise bounding box coordinates for 'Person' objects and calculating density maps.

Patterns to Avoid

Treating it like a general image editor

✗ AVOID

Trying to upload an image and asking the agent, 'Make this picture better' or 'Describe the mood.' The MCP is for structured data extraction, not creative edits.

✓ INSTEAD

To get structured data, use `analyze_image`. For example: 'Analyze this photo using EyePop.ai to list all detected objects and their labels.' This ensures you get coordinates and metadata.

Ignoring system status checks

✗ AVOID

Relying on a video analysis job without first verifying the connection, leading to silent failures or incomplete data sets.

✓ INSTEAD

Always start by using `check_eyepop_status`. This confirms your API is healthy before you try to run complex jobs like `analyze_video`.

Asking for a single, random piece of info

✗ AVOID

Just asking 'What are my Pops?' without context. The agent needs to know what kind of list you want.

✓ INSTEAD

Be specific. Ask: 'List all active visual pipelines (Pops) and retrieve the detection metadata for Pop ID X.' This guides the agent to use `list_pops` and then `get_pop`.

The Right Fit

Use this MCP if your core problem involves converting visual media (images, video) into actionable, structured data. You need to know *what* is in a picture or *how many* people walked past a certain point. It's perfect for security monitoring, retail analytics, and quality control where object detection matters.

Don't use it if your task is purely text-based (e.g., summarizing an article or classifying sentiment from a document). For pure data extraction from PDFs or documents, you need a different type of MCP designed for OCR or natural language processing. If all you need is simple cloud storage management, look for a file system connector instead.

Sifting through video footage is an absolute nightmare.

Right now, if you want to find out what happened during a specific time window—say, when merchandise was stolen or a safety protocol failed—you have to sit in dashboard after dashboard. You click on the timeline, zoom into the relevant frames, and then manually copy-paste coordinates or descriptions from separate reports just to build a narrative.

With this MCP, your agent does the heavy lifting. Instead of manual review, you ask it to process the stream. It handles the entire visual analysis pipeline, giving you clean JSON output with object labels, bounding boxes, and confidence scores—all delivered directly into your chat window.

EyePop.ai provides full-stack visual intelligence through the EyePop.ai MCP.

You no longer need to manually check connectivity or build new pipelines from scratch. You use `list_models` to select the perfect detection algorithm, then `create_pop` to set up a persistent monitoring job, and finally, you monitor its status using natural language queries.

The difference is control. You're not just viewing data; your agent is actively managing the entire life cycle of visual intelligence—from model selection to real-time detection reporting.

EyePop.ai: 10 Vision Analysis Tools

These tools let you programmatically perform every step of the computer vision process—from analyzing a single image to managing complex video streams.

#	TOOL	DESCRIPTION
01	`analyze_image`	Processes a single picture to return all detected objects, their descriptive labels, and precise coordinates.
02	`analyze_video`	Analyzes video files frame by frame, returning object detection results over the entire timeline.
03	`check_eyepop_status`	Confirms that your API connection to EyePop.ai is currently active and working.
04	`create_pop`	Builds a new, customized visual processing pipeline for recurring analysis tasks.
05	`get_account`	Retrieves general account information associated with your EyePop.ai credentials.
06	`get_model`	Fetches detailed information about a specific visual detection model you are using.
07	`get_pop`	Retrieves the current details and status of an existing visual pipeline (Pop).
08	`list_detections`	Lists all recorded object detections associated with a specific project or pop.
09	`list_models`	Provides an overview of all available pre-trained vision models you can use.
10	`list_pops`	Lists every active and inactive visual processing pipeline (Pop) in your account.

See It in Action

Real prompts you can use once this MCP is connected to your AI agent through Vinkius Cloud.

U List all active Pops in my EyePop.ai account.

I've retrieved your Pops. You currently have 5 active visual pipelines, including 'Main Security Feed' and 'Store Counter A'. Would you like the detailed detection metadata for any of them?

U Show the detected objects from 'Main Security Feed' for the last hour.

Visual intelligence orchestrated! For Main Security Feed, I've identified 10 object detections in the last hour, including 5 'Persons' and 2 'Vehicles'. I've retrieved the technical confidence metadata for your review. Need help setting an alert for specific objects?

U Check the processing status for Pop ID 'pop_123'.

Operational monitoring orchestrated! For pop_123, the current status is 'RUNNING' and processing 30 frames per second. Your API connection is healthy. Shall I retrieve the detailed resource usage metadata for this pipeline?

Frequently Asked Questions

01 **How do I analyze a single picture using the EyePop.ai MCP?**

You use the `analyze_image` tool. You simply prompt your agent and provide the image, specifying that you want object detection labels and bounding boxes returned.

02 **Can EyePop.ai track objects over time? Which tool do I use?**

Yes, use `analyze_video`. This tool processes video files to give you temporal results, tracking the same objects across multiple frames and showing their movement.

03 **I need recurring analysis. How do I set up a visual pipeline?**

You start by using `create_pop` to build your dedicated workflow. This establishes a persistent 'Pop' that you can then monitor later with `list_pops`.

04 **What if my API connection fails? How do I check it?**

You use the `check_eyepop_status` tool. This is the fastest way to confirm that your API key and account are communicating correctly with EyePop.ai.

05 **How many different types of models can I access via the MCP?**

You first call `list_models`. This tool provides a list of all available pre-trained vision algorithms, letting you choose the best fit for face recognition or object detection.

Go Live in 60 Seconds

Get your connection token from **cloud.vinkius.com**, then paste the endpoint URL into any MCP-compatible client.

YOUR MCP ENDPOINT

```
https://edge.vinkius.com/[TOKEN]/mcp
```

CLIENT	WHERE TO CONFIGURE
Claude AI	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
Cursor	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
VS Code	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add `"eyepopai": { "url": "…" }`
Windsurf	MCP Settings → `mcp_settings.json` → Add endpoint URL
ChatGPT	Settings → Tools & plugins → Add MCP server → Paste endpoint
Gemini	Extensions → Add MCP Server → Paste endpoint URL

ASK AN AI ABOUT THIS

Let your preferred AI explain this MCP server

- Ask ChatGPT
- Ask Claude
- Ask Perplexity
- Ask Gemini
- Ask Grok

READY TO CONNECT

EyePop.ai is live on Vinkius Cloud.

Get your connection token, paste it into your AI agent, and start building. No SDK. No deployment. Just results.

Start at cloud.vinkius.com →

vinkius.com · support@vinkius.com

INDEPENDENT PLATFORM DISCLAIMER

Vinkius is an independent platform and is not affiliated with, endorsed by, sponsored by, verified by, or otherwise authorized by EyePop.ai. All third-party trademarks, logos, and brand names are the property of their respective owners. Their use in this document is strictly for informational purposes to identify service compatibility and interoperability.

DOCUMENT INFORMATION

Generated	June 2026
MCP Server	EyePop.ai MCP
Server ID	`019dd0ed-f984-7039-9dbc-cd10ba82e5f9`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

LICENSE & USAGE

This document is generated automatically by the Vinkius PDF Engine. Content reflects the MCP server configuration at the time of generation and may change as updates are deployed. For the most current information, visit vinkius.com/mcp/eyepopai.