MCP SERVER | NO CODE | CLOUD HOSTED

Roboflow MCP

Manage the full CV lifecycle with natural prompts.

Roboflow manages your entire computer vision pipeline, letting you handle everything from dataset uploads to model training runs through natural language conversation. You can create new projects, download datasets in COCO or YOLO formats, and monitor metrics like mAP without leaving your agent.

A+ Quality Score 100/100

computer-vision | dataset-management | model-training | image-annotation | machine-learning | workflow-automation

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

SCAN TO VIEW

The connectivity layer between AI and the world's software.

Vinkius sits between AI and every application. All communication passes through Vinkius Cloud via the Model Context Protocol (MCP) — with governance, observability, and security at every layer.

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the connectivity layer where AI connects 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.

Roboflow MCP

29 tools available
Cloud-hosted on Vinkius

Managing a computer vision workflow used to mean jumping between six different dashboards: one for data annotation, another for version control, and a third just to check performance metrics. This MCP changes that. Your agent connects directly to Roboflow, letting you manage the full CV lifecycle using simple prompts. You can upload raw images via URL or Base64, track dataset health, and even fork public projects straight into your private workspace. Whether you're an ML engineer monitoring a training run or a data scientist needing specific versions of data for custom scripts, this tool makes it happen in plain conversation. This capability is available through the Vinkius catalog, giving you one connection point to hundreds of other services. You start by setting up projects and then use your agent to run inference on images right away.

Core Capabilities

01 — Build and Organize Projects

Create new project folders or fork public Roboflow Universe projects into your private workspace.

02 — Manage Image Assets and Data Versions

Upload images using URLs, manage dataset versions, check class distribution health, and download the data in required formats like COCO or YOLO.

03 — Train and Monitor Models

Start model training runs on specific dataset versions and retrieve detailed performance metrics, including mAP, precision, and recall.

04 — Search and Verify Data

Filter images within a project or an entire workspace to audit data quality or run real-time inference against hosted models.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Roboflow Private API Key.

02 Your agent uses the key to authenticate and retrieve your default workspace details.

03 You prompt your agent with a command, like 'Train a model on version 3,' and the tool executes the job and returns the status.

The bottom line is you manage complex visual data workflows without ever touching a dashboard or API call yourself.

Built For

This MCP targets anyone whose job involves building, testing, or scaling machine learning models that rely on image recognition. If your current process requires jumping between data storage, version control, and training dashboards, you need this.

Machine Learning Engineer

Monitors the progress of model training runs and retrieves precise performance metrics without needing to leave their IDE or terminal.

Data Scientist

Quickly queries specific dataset versions and exports data for custom, external training scripts using natural language prompts.

Product Manager

Audits model performance metrics and visualizes inference results through simple conversation to validate product features.

What Changes When You Connect

01 Speed up data prep by running `auto_label` jobs; your agent handles assigning foundational models to label images, saving manual annotation time. You'll get labeled assets ready for training almost instantly.

02 When you need to audit your data, use the built-in search tools like `search_workspace_images`. Instead of manually clicking through thousands of files, your agent pulls up exactly what you're looking for based on tags or metadata.

03 Stop guessing if your model works. Run an instant test using `run_inference` to check how a hosted model behaves against new images right in the conversation window. It's immediate feedback without setup.

04 Data versioning is simplified: Use `get_version` and then `start_training`. You tell your agent which dataset version you want, and it handles pointing the training job to the correct data snapshot.

05 Organization becomes simple. Need a new project? Just run `create_project`. If you find a good public example, use `fork_universe_project` to start from that baseline instead of building from scratch.

Real-World Applications

Validating Model Performance Post-Deployment

A product manager needs to know if the model can detect 'hard hats' on images taken in a new warehouse environment. They ask their agent, and it uses `run_inference` against a few test images, instantly providing visual confirmation of false positives or negatives.

Expanding Dataset Scope for New Classes

A data scientist realizes the 'Glove Compliance' model is weak on dirty gloves. They use `upload_image` to add a batch of new photos and then run `get_dataset_health`. The agent reports low class distribution metrics, telling them exactly where the dataset needs more focus.

Quickly Setting Up an Enterprise Project

An ML engineer joins a new team. Instead of starting from scratch, they use `fork_universe_project` to copy a proven 'industrial safety' template into the workspace. They then run `create_folder` and organize their assets immediately.

Retraining After Data Changes

A team manually corrects hundreds of annotations, which are uploaded via `upload_annotation`. Now they need to retrain. The agent uses `get_version` to lock the new data state and then runs `start_training`, giving them a predictable path from cleanup to deployment.

Patterns to Avoid

Manually tracking training status

✗ AVOID

Having to log into the Roboflow dashboard, navigate to the 'Training' tab, and refresh the page every five minutes just to see if the job is stuck or finished.

✓ INSTEAD

Ask your agent to `get_training_results` once the job ID is confirmed. The MCP pulls all current metrics and status into the chat window instantly.

Forgetting data formats

✗ AVOID

Downloading a dataset ZIP file only to realize the format isn't compatible with your custom PyTorch script, requiring manual conversion or re-downloading.

✓ INSTEAD

Use `download_dataset` and specify the required output format (e.g., COCO). The MCP handles retrieving the data in the exact structure you need for scripting.

Overwriting critical versions

✗ AVOID

Accidentally deleting a perfectly annotated dataset version because it was only visible in a nested folder structure, resulting in lost training history.

✓ INSTEAD

Always use `get_project` to view the full metadata and then check the trash using `list_trash`. If needed, run `restore_trash` before moving on.

The Right Fit

Use this MCP if your primary workflow involves iterative computer vision development: collecting raw images, annotating them, versioning the dataset, training a model, and testing it. It handles the entire loop from data ingress to performance reporting. Don't use this if you only need simple file storage or basic image tagging; those tasks are better handled by dedicated asset management tools. If your goal is purely to build complex, multi-step workflows that involve external APIs (like sending emails or interacting with a CRM), you might prefer an MCP focused on communication protocols rather than data science pipelines.

The headache of managing visual data assets.

Today, getting your model ready involves a mess of clicks. You upload raw images to one place, use another tool for annotation, and then you have to manually track which version of the dataset was used for training—all while jumping between five different browser tabs just to check if the job finished.

With this MCP, that process collapses into conversation. Your agent manages the entire visual data lifecycle. You tell it what to do—whether it's uploading a batch of images via URL or checking the class distribution using `get_dataset_health`. The result is immediate: you get clear status updates and actionable results right where you are.

Roboflow MCP delivers full dataset control.

You eliminate the need to manually manage versions or check data integrity. You no longer have to hunt for a specific annotated asset; your agent handles it using tools like `get_version` and `list_workspace_projects`. The complexity of CV pipelines is hidden behind simple commands.

The difference now is that you're not just running a tool; you're managing an entire, integrated pipeline. You gain control over every asset, every version, and the performance metrics in one single interaction.

Roboflow: 29 Tools for Computer Vision

These tools let you manage every step of a computer vision project, allowing your agent to organize projects, upload assets, train models, and verify data quality using natural language.

#	TOOL	DESCRIPTION
01	add_projects_to_folder	Adds existing projects into a designated folder within an enterprise workspace.
02	auto_label	Starts an automated labeling job using foundation models to speed up annotation.
03	cancel_training	Stops a model training run that is currently active, saving computational resources.
04	create_annotation_job	Assigns a batch of images to specific labelers and reviewers for human annotation work.
05	create_folder	Creates new project folders within an enterprise workspace structure.
06	create_project	Sets up a brand new, empty machine learning project in your account.
07	delete_images	Removes multiple images from a specific project folder.
08	delete_project	Deletes an entire project or dataset version, moving it to the trash bin.
09	download_dataset	Generates a download link for all images in a zipped file format (COCO, YOLO, etc.).
10	fork_universe_project	Copies an existing public project from Roboflow Universe to your private account.
11	get_async_task	Tracks the status of long-running background operations, like large exports or forks.
12	get_dataset_health	Checks the structural integrity of a dataset, looking at class balance and missing annotations.
13	get_image	Retrieves specific metadata details for one image file in your project.
14	get_project	Fetches comprehensive details and version history for a given project.
15	get_root	Verifies the connection credentials and retrieves the default workspace name.

#	TOOL	DESCRIPTION
16	`get_training_results`	Pulls metrics and status updates for a specific, completed training run version.
17	`get_version`	Gets metadata details about a particular version of your dataset.
18	`list_folders`	Lists all the project folders available within an enterprise workspace.
19	`list_trash`	Displays a list of items that have been deleted and moved to trash.
20	`list_workspace_projects`	Lists all projects associated with the current workspace account.
21	`manage_image_tags`	Adds, removes, or sets descriptive tags on one or more images.
22	`restore_trash`	Brings a deleted item back from the trash bin into active use.
23	`run_inference`	Runs an immediate test on an image using pre-hosted models to see its predicted output.
24	`search_project_images`	Finds and filters images based on criteria within a single project.
25	`search_workspace_images`	Searches for images across the entire workspace using defined filters.
26	`start_training`	Initiates a model training job on a specified dataset version.
27	`stop_training`	Stops an active model training process early if it's not performing well enough.
28	`upload_annotation`	Attaches a file containing annotations to an existing image asset.
29	`upload_image`	Uploads new images directly into a specified project folder.

See It in Action

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

U List all projects in my Roboflow workspace 'industrial-safety'.

I've retrieved the projects for 'industrial-safety'. You have 3 active projects: 'Hard Hat Detection' (object-detection), 'Glove Compliance' (classification), and 'Forklift Tracking'. Which one would you like to inspect?

U Upload this image URL to the 'Hard Hat Detection' project in workspace 'industrial-safety'.

Uploading image... Success! The image has been added to the 'Hard Hat Detection' project. It is currently in the unassigned batch. Would you like to assign it to a specific split?

U Show me the training metrics for version 5 of the 'Forklift Tracking' project.

Fetching results for version 5... The model achieved a mAP of 88.5%, with a precision of 91.2% and recall of 84.7%. The training run is completed. Would you like to see the full metrics breakdown?

Frequently Asked Questions

01 **How does Roboflow MCP handle large datasets?**

It handles them by providing tools like `get_dataset_health` to audit class distribution across massive uploads and allowing you to download structured data via `download_dataset` for external use.

02 **Can I train a model using Roboflow MCP?**

Yes, your agent initiates training runs by calling `start_training`. It then lets you monitor the process and retrieve metrics using `get_training_results` until it's complete.

03 **What if I need to improve my model with new images?**

You can use `upload_image` or `search_workspace_images` to find and add the necessary assets. After adding them, you run `get_dataset_health` again to ensure the balance is correct before retraining.

04 **Does Roboflow MCP help with project organization?**

Absolutely. You can use `create_project` or `add_projects_to_folder` to structure your work, and even clone public examples using the `fork_universe_project` tool.

05 **How do I test my model without running a full training job?**

You can run an immediate prediction check by calling `run_inference`. This tests your existing models on new images and gives you real-time results instantly.

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 `"roboflow": { "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

Roboflow 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 Roboflow. 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	Roboflow MCP
Server ID	`019e38e5-62d8-7158-a3f9-2e42ac969dec`
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/roboflow.