AI CAMERA HTTP SERVER

User Guide

AutoNexIOT

www.autonexiot.com

Version 1.0

Contents

AI CAMERA HTTP SERVER....1
1. Introduction....2
2. System Requirements....3
3. Application Overview....4
4. Quick Start....7
5. Modbus TCP Server....11
6. Object Class Index....12
7. Program Activation....15
8. Troubleshooting....16
9. Support and Contact....17

1. Introduction

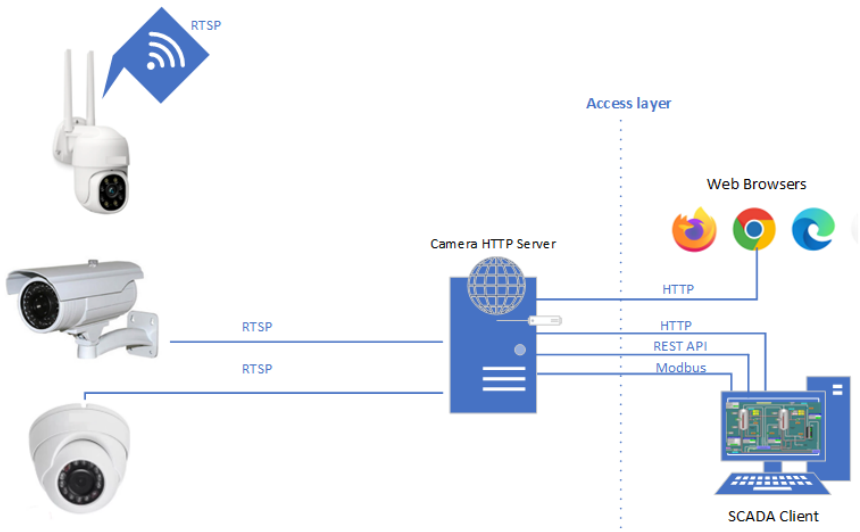
AI Camera HTTP Server converts RTSP camera streams into standard HTTP video feeds that any web browser or SCADA application can display. It also runs an AI object detection model against each stream in real time, making the detection results available through a REST API and a Modbus TCP server.

The typical deployment looks like this: IP cameras send RTSP streams to the server. The server re-serves each stream as an HTTP MJPEG feed on a dedicated port. Simultaneously, the AI engine analyses the video and counts detected objects by class. Those counts are exposed over HTTP (for web and SCADA clients) and over Modbus TCP (for PLC and DCS systems that read holding registers directly).

Key capabilities:

- RTSP-to-HTTP stream conversion for any web browser or SCADA HMI.
- Real-time AI object detection across 80 COCO object classes.
- REST API for JSON-based integration with modern applications.
- Modbus TCP server for direct PLC and DCS integration.
- System tray operation — the server continues running when the window is closed.

Figure 1 — System architecture overview



2. System Requirements

The following minimum specifications apply. GPU memory is the primary constraint when running multiple camera streams simultaneously — each stream adds load. For deployments with eight or more cameras, 16 GB or more of VRAM is recommended.

Category	Requirement
Operating System	Windows 10 (64-bit)
Operating System	Windows 11 (64-bit)
Operating System	Windows Server 2019
Operating System	Windows Server 2022
GPU	4 GB VRAM minimum (more recommended for multiple cameras)
RAM	8 GB minimum
CPU	2.4 GHz, 4 cores minimum
GPU Driver	DirectX 12 feature level 12_0 or higher
.NET Runtime	.NET 8.0 Desktop Runtime (v8.0.8) x64

Note: The AI detection engine uses DirectML and runs on any DirectX 12-compatible GPU — NVIDIA, AMD, or Intel. There is no requirement to install CUDA, cuDNN, or any NVIDIA-specific packages. If no compatible GPU is found, the engine automatically falls back to CPU processing.

.NET Runtime Installation

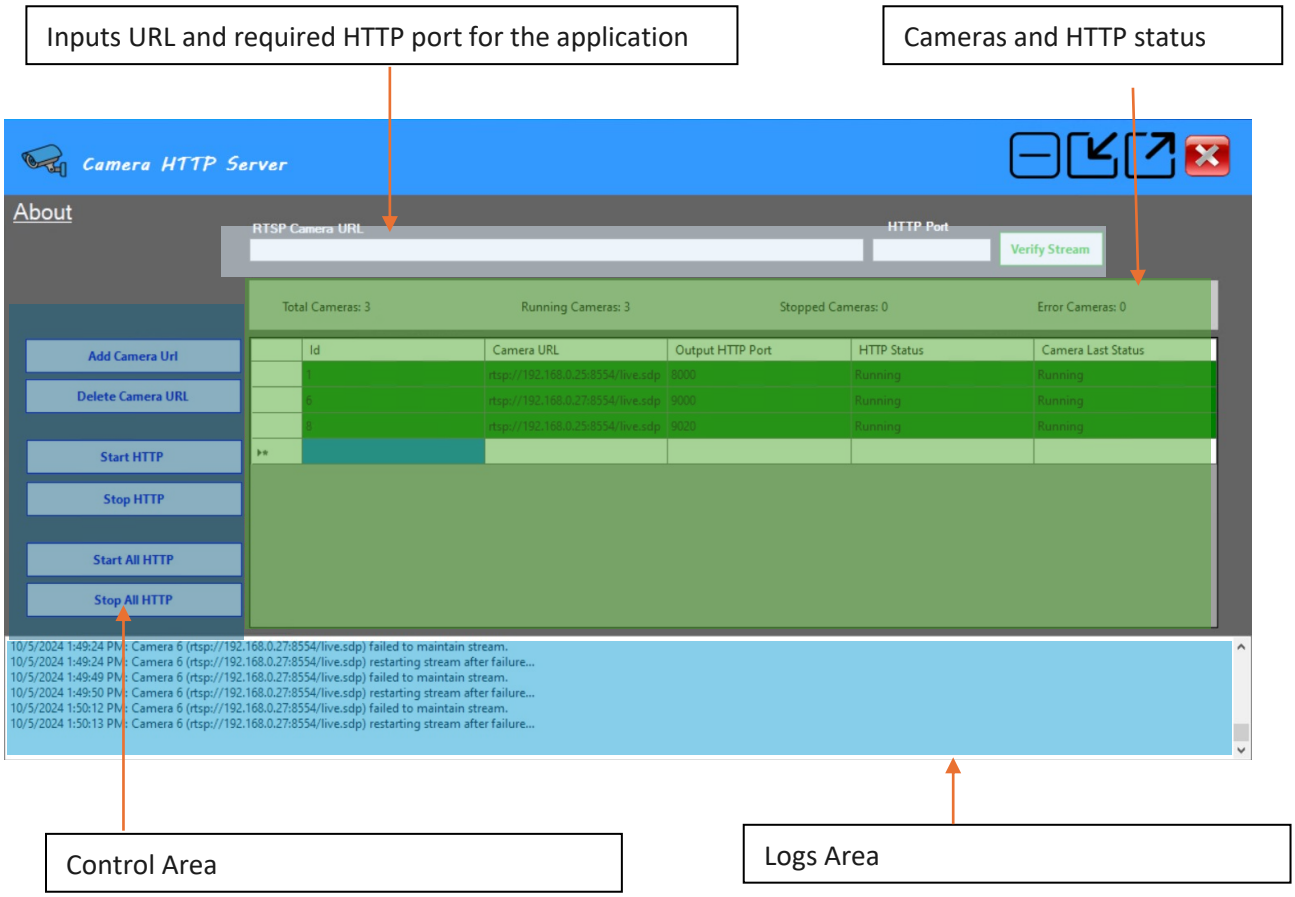
Download and install .NET 8.0 Desktop Runtime (v8.0.8) x64 before running the application. The installer is available at:

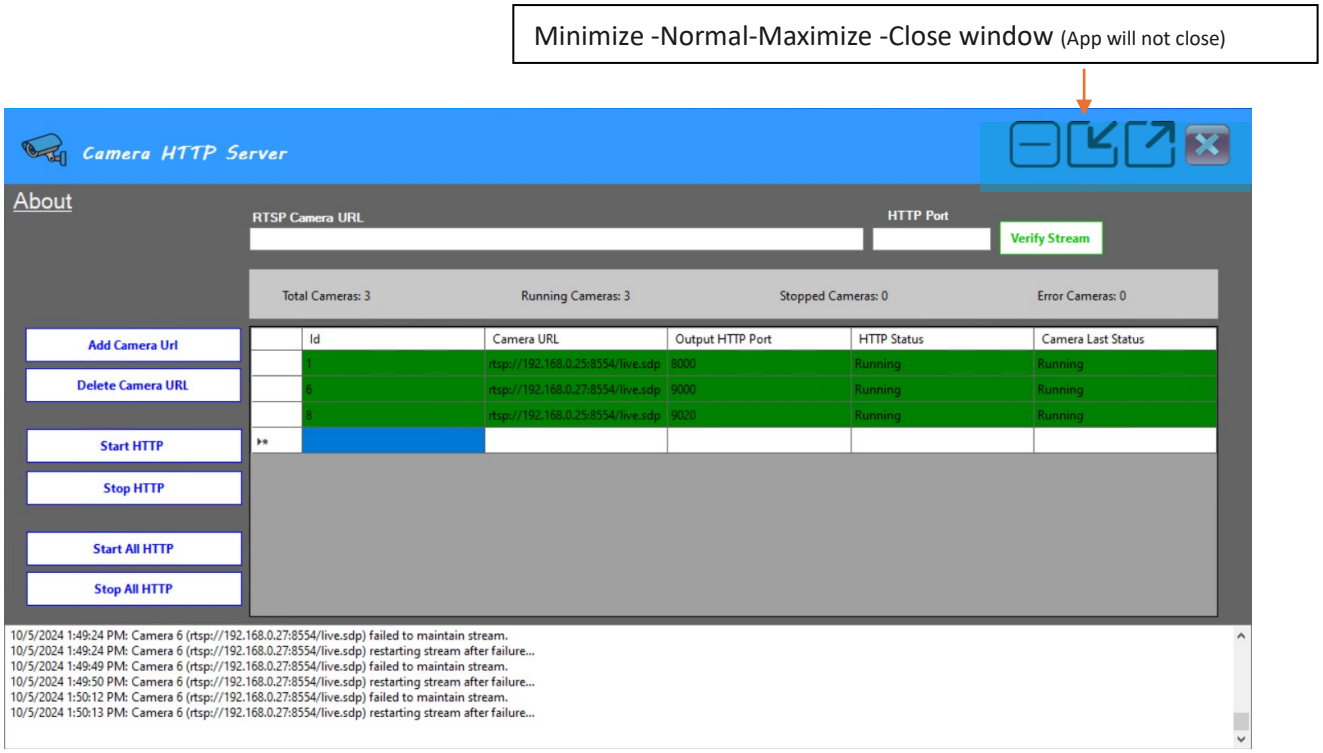
https://dotnet.microsoft.com/en-us/download/dotnet/thank-you/runtime-desktop-8.0.8-windows-x64-installer

3. Application Overview

The main window is divided into four functional zones. Understanding each one makes day-to-day operation straightforward.

Figure 2 — Main application window





Control Area (left panel)

Contains the buttons for adding and removing camera entries, and for starting or stopping individual or all HTTP streams. Select a row in the camera grid first, then use Start HTTP or Stop HTTP to control that specific camera. Start All HTTP and Stop All HTTP operate on every camera simultaneously.

Input Bar

The RTSP Camera URL field accepts the stream address of the camera. The HTTP Port field sets the port this camera will be served on. The Verify Stream button tests the RTSP connection before you commit to starting the stream.

Camera Status Grid

Each row represents one camera. The grid shows the camera URL, assigned HTTP port, HTTP server status (Running or Stopped), and the live camera connection status. Rows turn green when the stream is active and healthy.

Log Area

A scrolling event log covering connection attempts, reconnections, detection session initialisation, GPU or CPU provider selection, and any errors. This is the first place to check when diagnosing an issue.

Note: Closing the main window does not stop the server. The application minimises to the system tray (notification area). Right-click the tray icon and select Show to restore the window, or Exit to fully shut down the application.



Figure 3 — System tray icon

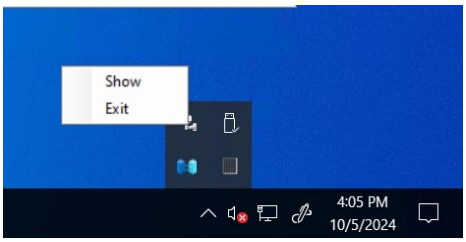


Figure 4 — Tray right-click menu

4. Quick Start

Follow the steps below in order. Getting a camera live typically takes two to three minutes.

1. Add the RTSP Camera URL

Type or paste the camera's RTSP address into the RTSP Camera URL field. The format is typically:

```
rtsp://<username>:<password>@<camera-ip>:554/live.sdp
```

The exact path after the port varies by manufacturer. Check your camera's web interface or documentation if you are unsure. Then click Add Camera URL to save it.

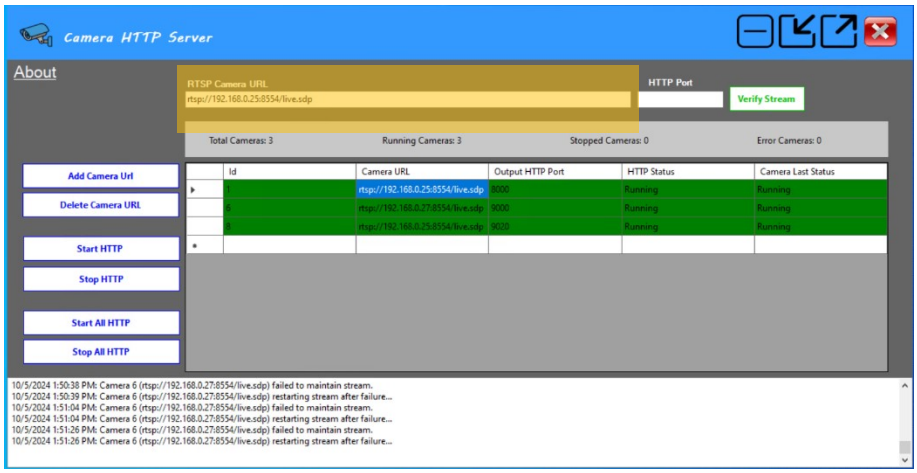
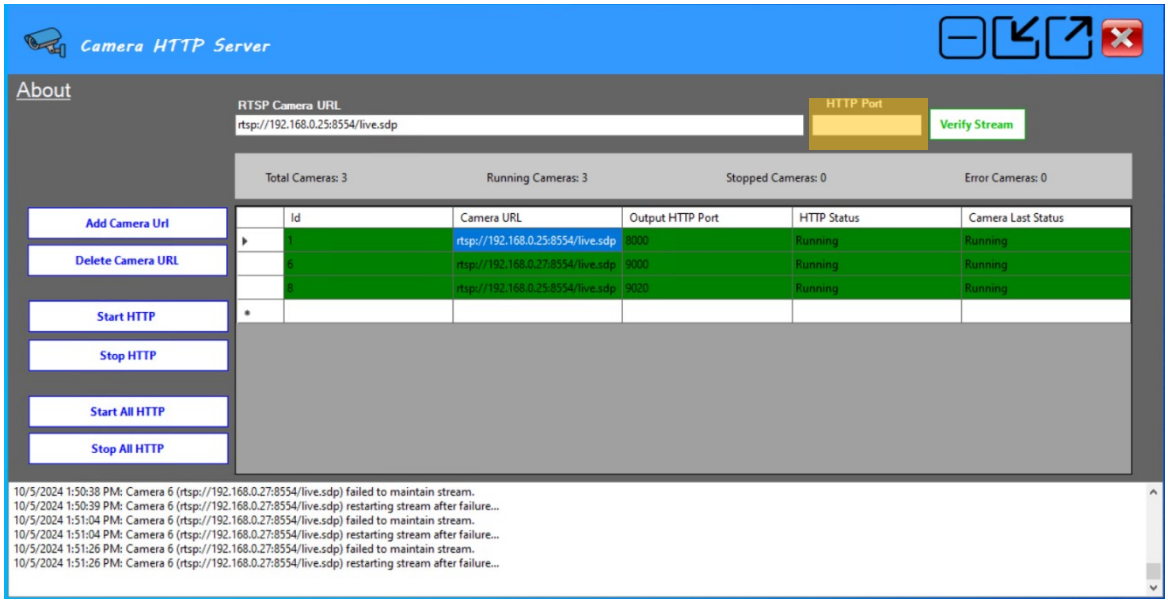


Figure 5 — Adding a camera URL

2. Assign an HTTP Output Port

Enter a port number in the HTTP Port field. Ports from 8080 upward work well. Each camera must have a unique port — assigning the same port to two cameras will prevent the second one from starting. Ensure the port is open in Windows Firewall.

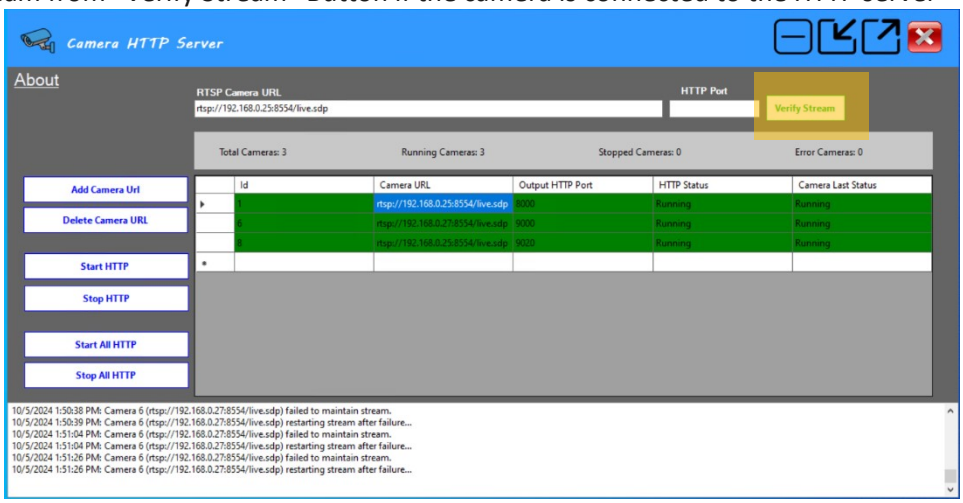


3. Verify the Stream

Click Verify Stream to test the RTSP connection. The Camera Last Status column will show Running if the camera is reachable, or Error if there is a connectivity problem.

Troubleshooting: If Verify Stream returns an error, confirm the camera IP is pingable from the server, that RTSP port 554 is not blocked by a firewall, and that the username and password in the URL are correct.

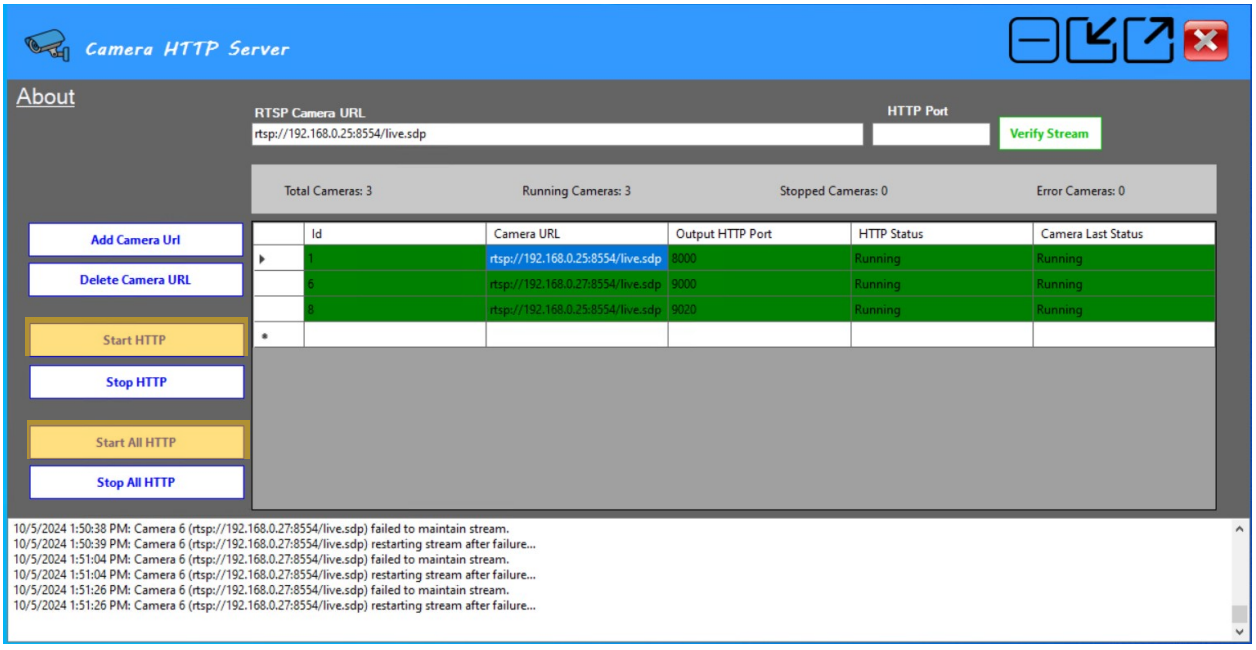
Verify the stream from “Verify Stream” Button if the camera is connected to the HTTP server



4. Start the HTTP Stream

Select the camera row in the grid and click Start HTTP. The HTTP Status column changes to Running and the row turns green. To start all cameras at once, click Start All HTTP.

- select the camera URL row and start the streaming.



5. Access the Video Stream

Open a browser and navigate to the server's IP address and the port you assigned:

`http://<server-ip>:<port>`

The stream can be embedded directly in a SCADA graphics page using a web browser control pointed at this URL.

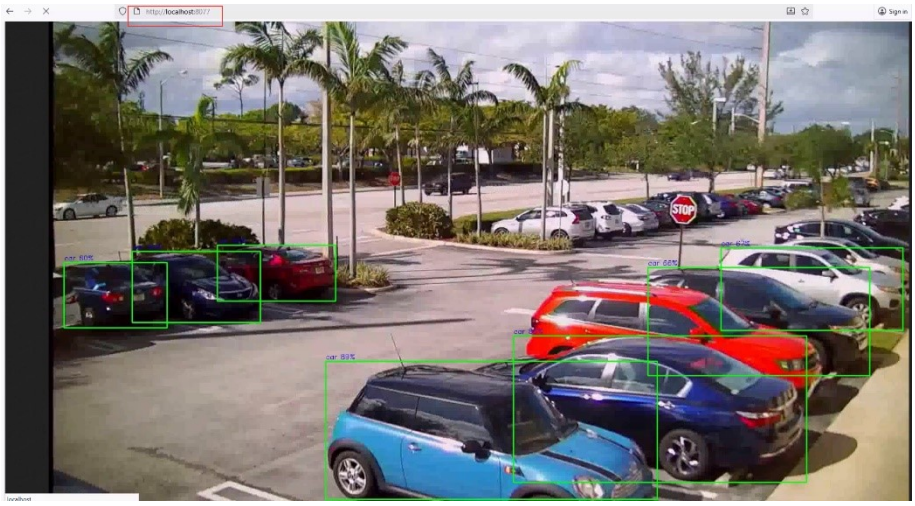


Figure 6 — Live stream with AI detection overlays visible in a browser

6. Access Object Detection Counts via REST API

Query the object counts for any camera using an HTTP GET request:

`http://localhost:8081/api/counts/<camera-id>`

Replace <camera-id> with the numeric ID shown in the first column of the camera grid. The response is a JSON object listing each detected object type and its current count.

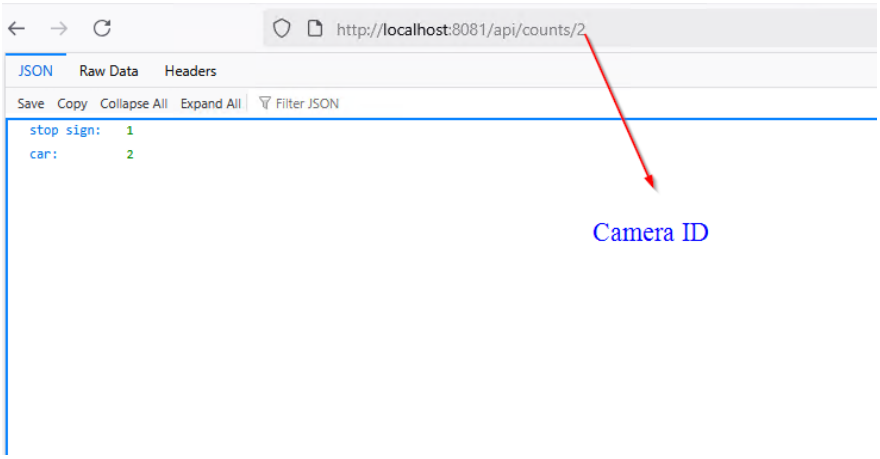


Figure 7 — REST API JSON response in a browser

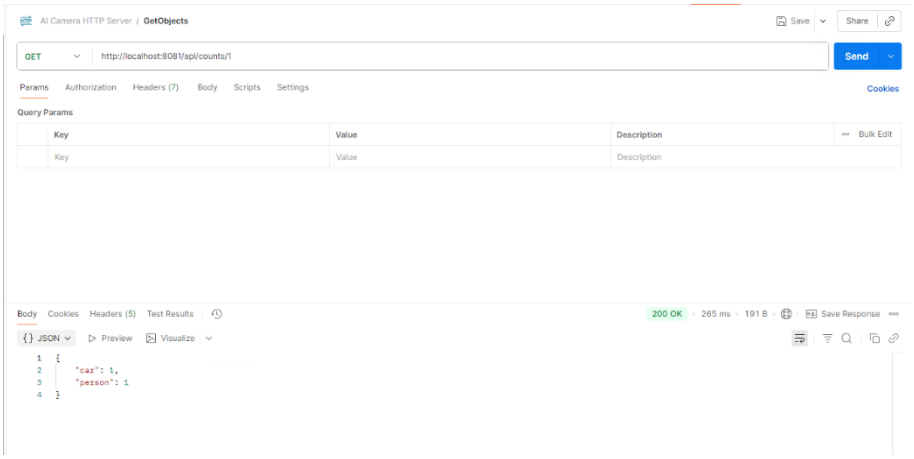


Figure 8 — Application with a camera running and REST API label visible in postman

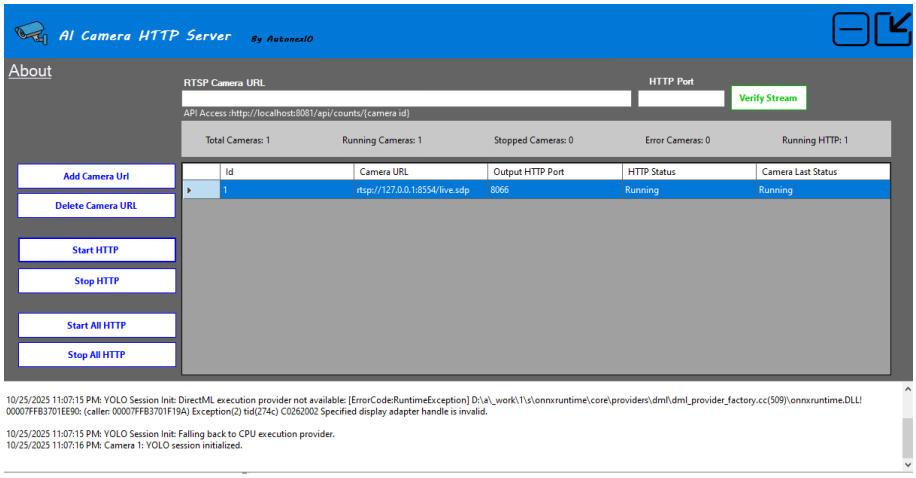


Figure 9 — Camera status should be running

5. Modbus TCP Server

The same detection data exposed through the REST API is also available via a built-in Modbus TCP server running on port 502. This allows PLC and DCS systems to read object counts directly as holding register values, without any scripting or middleware.

Each camera is allocated a block of 100 holding registers. Within that block, each register position corresponds to a specific object class (see Section 6 for the full index). The register value is the count of that object currently detected in the camera's scene, updated continuously while the stream is running.

Register Address Mapping

Camera ID	Register Range	Notes
Camera 1	40001 – 40100	Registers for all 80 object classes
Camera 2	40101 – 40200	Registers for all 80 object classes
Camera N	40001 + (N–1) × 100 to 40100 + (N–1) × 100	Pattern continues for each camera

Important: When a camera stream has failed or been stopped, all 100 of its registers are set to –1 (which reads as 65535 in unsigned 16-bit Modbus). Your PLC logic should check for this value and treat it as a fault condition rather than a valid object count.

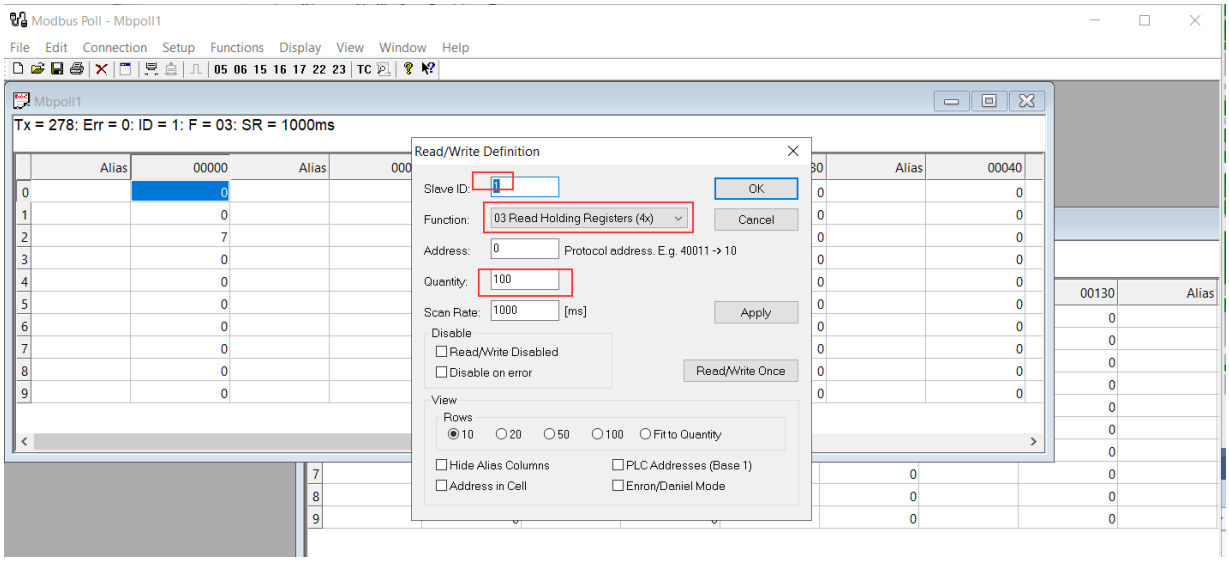


Figure 10 — Reading holding registers in a Modbus client

6. Object Class Index

The detection model recognises 80 object classes drawn from the COCO dataset. The table below lists each class alongside its register index. Within each camera's 100-register block, the index is the offset from the camera's base address.

For example, to read the **car** count (index 2) for **Camera 1**, read holding register `40003`. For Camera 2, read register `40103`.

Index	Object Type
0	person
1	bicycle
2	car
3	motorbike
4	aeroplane
5	bus
6	train
7	truck
8	boat
9	traffic light
10	fire hydrant
11	stop sign
12	parking meter
13	bench
14	bird
15	cat
16	dog
17	horse
18	sheep
19	cow
20	elephant
21	bear
22	zebra
23	giraffe
24	backpack
25	umbrella

Index	Object Type
26	handbag
27	tie
28	suitcase
29	frisbee
30	skis
31	snowboard
32	sports ball
33	kite
34	baseball bat
35	baseball glove
36	skateboard
37	surfboard
38	tennis racket
39	bottle
40	wine glass
41	cup
42	fork
43	knife
44	spoon
45	bowl
46	banana
47	apple
48	sandwich
49	orange
50	broccoli
51	carrot
52	hot dog
53	pizza
54	donut
55	cake
56	chair
57	sofa
58	potted plant

Index	Object Type
59	bed
60	dining table
61	toilet
62	tv monitor
63	laptop
64	mouse
65	remote
66	keyboard
67	cell phone
68	microwave
69	oven
70	toaster
71	sink
72	refrigerator
73	book
74	clock
75	vase
76	scissors
77	teddy bear
78	hair drier
79	toothbrush

7. Program Activation

The unactivated version supports up to three cameras. To add more, you need a licence tied to your machine's hardware ID.

Activation Procedure

7. **Open the About window**

   Click About in the top-left corner of the application.

8. **Copy your Machine ID**

   The Machine ID is displayed in the yellow field. This ID is unique to the current machine and cannot be transferred to another system.

9. **Request an activation code**

   Email the Machine ID to Autonexiot, specifying the number of cameras you require:

   sales@Autonexiot.com

10. **Enter the activation code**

    Paste the code you receive into the Activation Code field and click Activate. A confirmation dialog will appear. Click OK — the camera limit is lifted immediately.

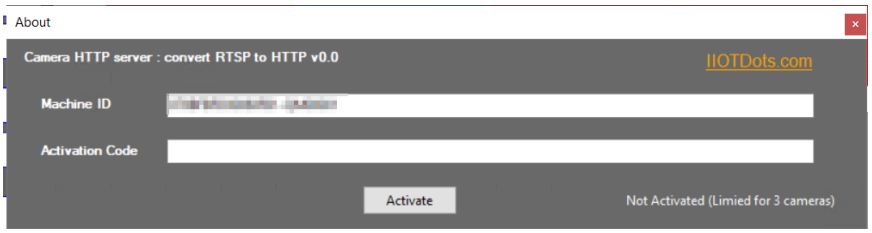


Figure 11 — About window showing Machine ID (not yet activated)

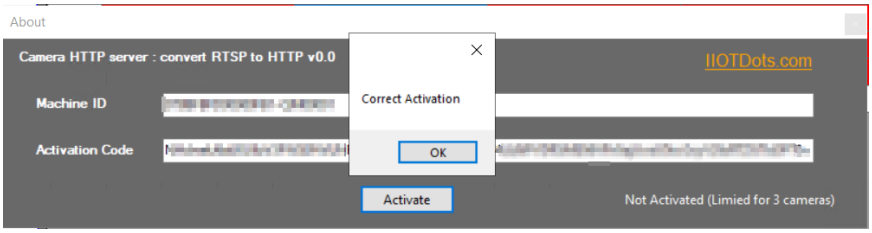


Figure 12 — Successful activation confirmation

Note: The Machine ID is hardware-bound. If you reinstall Windows or significantly change the hardware on the server, contact Autonexiot to request a replacement activation code.

8. Troubleshooting

The table below covers the most common issues encountered during installation and operation.

Symptom	Likely Cause and Action
Camera shows 'Error' status	The RTSP URL is unreachable. Verify the camera IP is pingable, that RTSP port 554 is not blocked by a firewall, and that the credentials in the URL are correct.
HTTP port Running but browser cannot connect	Windows Firewall is blocking the inbound port. Add an inbound rule for the HTTP port, or verify no other service is using it.
Detection session falling back to CPU	The GPU driver does not support DirectX 12 feature level 12_0, or the machine is a VM without GPU passthrough. Update the driver and check DirectX support via dxdiag.
Modbus registers reading 65535 (0xFFFF)	This is the camera error state — all registers are set to –1 signed when the stream is stopped or has failed. Diagnose the camera connection then restart the HTTP stream.
REST API returns empty or stale counts	The detection model runs every few frames. Wait a few seconds after starting a new stream. If counts remain empty, check the log to confirm the detection session initialised successfully.
Stream reconnects repeatedly	The camera's RTSP feed is dropping. Common on WiFi cameras or those with unstable power. The application retries automatically. Check the camera's network connection and consider reducing the requested resolution.
Port conflict on startup	Two cameras have been assigned the same HTTP port. Each camera requires a unique port number.

9. Support and Contact

For licensing enquiries, activation codes, and technical support not covered in this guide, contact the Autonexiot team directly.

Sales & Licensing	sales@Autonexiot.com
Product Website	Autonexiot.com

When requesting an activation code, include your Machine ID (visible in the About window) and the number of camera streams you need to license. This avoids back-and-forth and allows the code to be issued promptly.