engiAI Ethernet System

ZONE-PAIR — visual inspection of printed circuit boards and assemblies

The engiAI interface is available in English: button and field names in this manual are given exactly as they appear in the program.

> **The core principle of the inspection is "zone-pair inspection."** The circuit board is divided into zones, and each zone requires a specific inspection. Each model maintains a pair of "normal/anomaly" classes for each zone and only assesses within that pair. In the example provided, there are two zones: the C30 capacitor and the BR1 bridge; there could be hundreds of such zones on a real circuit board, and the method is the same.
>
> **Synthetic data – automatic shifting of the component from the base.** The program takes the component along with the surrounding area of the PCB (lines, tracks, base color), reduces the area to the size of the component, and shifts or rotates it. Defects for training are created in two clicks, without needing a single defective PCB.

User Manual: How to connect the camera, define zones, specify the alignment region, capture normal and anomalous data, train the model, initiate the inspection, and link it to the rejector: using audio alerts, discrete outputs from SICK and HIKROBOT cameras, and TCP communication with the robot or PLC.

Step 1	**Step 2**	**Step 3**	**Step 4**	**Step 5**	**Step 6**
Camera	Zone Grid	Normal vs. Anomalies	Training	Inspection	Exits and TCP

All screenshots were taken from the working program: vtlift software, 3264×2448 USB camera, model model_4 for two zones of 256×256 (C30 capacitor and BR1 bridge), trained in grayscale. The program works with HIKROBOT (GigE and USB3) cameras, SICK picoCam (uEye), USB/webcams, and IP cameras via RTSP. Version 3 – September 2026.

Contents

1. Why engiAI and the ZONE-PAIR method?
    - 1.1. What advantages does this method offer compared to classifying the entire frame?
    - 1.2. Speed: The AI only focuses on the relevant excerpts.
    - 1.3. What level of accuracy can be expected: calculating based on our camera
    - 1.4. Quick Start: Settings are pre-configured, and the zones and model can be set up in just a few clicks.
2. Learn the program interface in one minute
3. Cameras: Connection and Setup
    - 3.1. Which cameras are supported?
    - 3.2. Camera Window: Add, Reconfigure, or Remove
    - 3.3. Camera tab for HIKROBOT and SICK
    - 3.4. Camera Tab for USB/Webcam
    - 3.5. Saved Camera Settings (Presets)
    - 3.6. GigE Network and Reconnection
4. The ZONE-PAIR Model: From Grid to Trained Model
    - 4.1. How the method works: one model, one class per zone
    - 4.2. Develop a model
    - 4.3. Mark payment areas
    - 4.4. Record the reference: alignment region
    - 4.5. Zone-based image: normal and its synthetic data
    - 4.6. Synthetic data defect: The element shifts along with the base with just two clicks.
    - 4.7. Anomalies in Zones: Settings Window
    - 4.8. References to zones: solder pads and zone editor
    - 4.9. Model Training
5. Inspection
    - 5.1. Assign a model and verify the zones.
    - 5.2. Zone Recognition Threshold
    - 5.3. The "Live Conveyor": Payment in Any Position
    - 5.4. Which signal should be analyzed?
    - 5.5. Yellow Zones: Turn off the zone by clicking
    - 5.6. What you can see on the screen and in the console
6. Triggers: External sensor on the camera's input
7. Outputs: rejector, discrete outputs for cameras, and TCP
    - 7.1. Reference: Outputs: Contacts and Tester Verification
    - 7.2. OUTPUT Window: Rule for the Zone
    - 7.3. Sending a TCP packet to the robot or PLC
    - 7.4. Discrete camera output
    - 7.5. Audio alerts: a unique sound for each zone
    - 7.6. Log of activations and database

8. Other modes and tasks

9. Settings, console, requirements, and license

10. A concise checklist and troubleshooting guide

How to use the manual: Section 1 explains the purpose of the method and the expected results. Sections 3-5 outline the steps from a blank program to a working inspection, and should be followed in order. Sections 6-7 cover sensor, output, and TCP connections. Sections 8-10 cover the remaining features, settings, and a checklist.

1. Why engiAI and the ZONE-PAIR method?

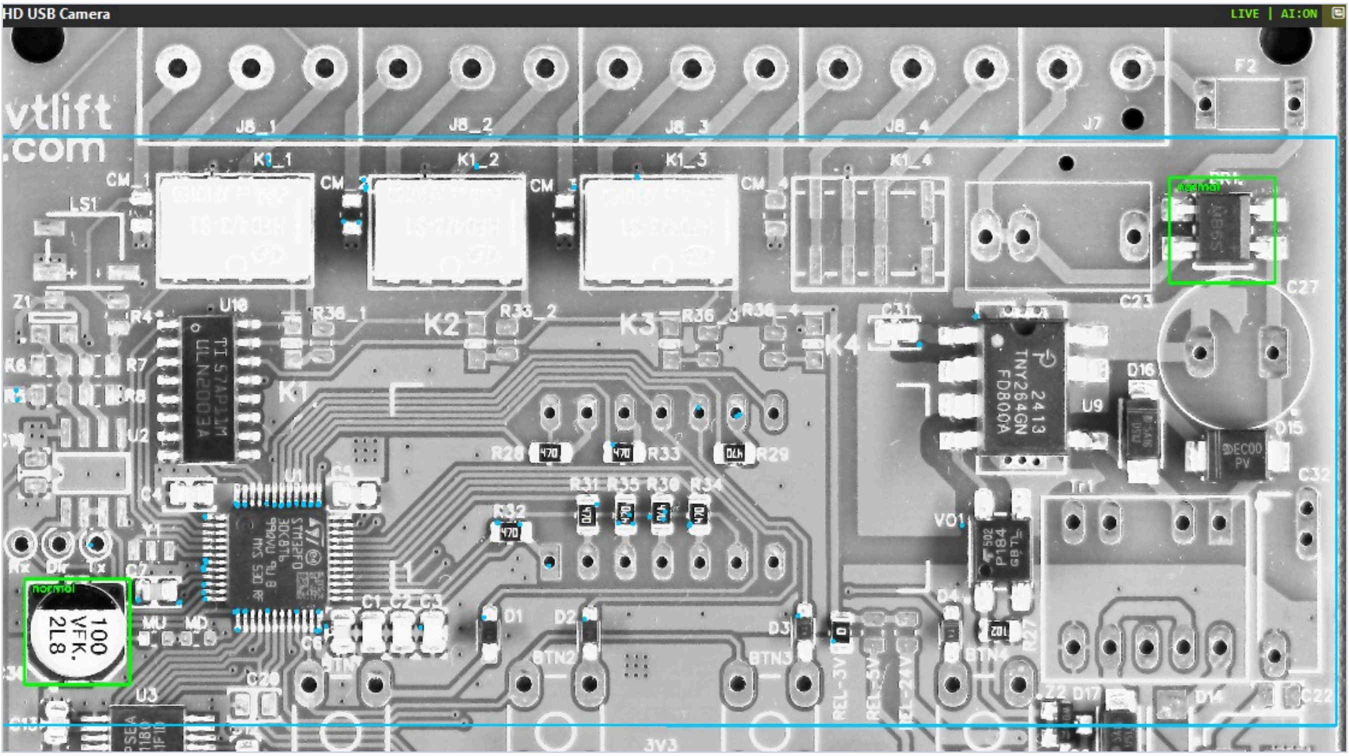
engiAI Ethernet System – a visual inspection program for production lines. It takes a frame from an industrial or standard camera, divides it into zones, checks each zone using a trained neural network, and converts the result into an action: a signal to the camera's discrete output, a TCP message to a robot or PLC, or a record in a log. The ZONE-PAIR method is designed for products where each location has its "correct" component: printed circuit boards, assemblies, sets of parts in a tray, and packaging with specific layouts.

1 model	2 zones	≈5 px	2 clicks
for all zones on the payment card, instead of a hundred separate models	As an example, the C30 capacitor and the BR1 bridge. There could be hundreds of these components on a circuit board.	The offset of the element to which the "normal/defect" boundary is applied.	From live frames to normal and synthetic defect frames for training

1.1. What does the method offer compared to classifying the entire frame?

A common approach is to use a single model with two classes, "normal/anomaly," applied to the entire image or to specific zones. However, this doesn't work on the circuit board: a gap under the BR1 bridge in zone 2 indicates a defect, while the same gap in another area of the board is considered normal. A single model cannot distinguish between these two cases. The other extreme is to use a separate model for each component, which would require dozens of training runs and numerous files in memory.

ZONE-PAIR addresses this by using a single model, with as many classes as there are zones. A "**Zone**" is a rectangle surrounding a single component. A "**Class**" is a folder containing images that the network learns from. Each zone has its own pair: "`zone_001`" (normal zone 1, for example, the C30 capacitor) and "`zone_001_anomaly`" (defect zone 1); "`zone_002`" and "`zone_002_anomaly`" represent the BR1 bridge. When checking zone 1, the program only looks at its pair; the answers for zone 2 are not used. Anomaly classes are optional: the "defect" changes the appearance of the zone, and the confidence in the "normal" state decreases automatically; however, with synthetic data (section 4.6), they can be created with a single click and noticeably improve reliability.



Live frame from the example: two zones – the BR1 bridge on the top right and the C30 capacitor on the bottom left, both green with the label "normal". The blue lines are the alignment region boundaries, and the blue dots are the reference points used to locate the board in the frame.

Decision within a pair

When checking zone 17, only two outputs from the network are considered: "normal zone 17" and "anomaly zone 17". The remaining hundreds of classes are not involved, so similar adjacent zones do not affect the result.

A foreign signal will not be recognized as a normal signal.

If a pair of adjacent zones has a network coverage of less than 10%, the zone is considered "unrecognized": this indicates a different signal or a shifted network. Such a zone is considered a defect, not a "normal" one with a high probability.

Defect size is proportional to the area

The size and position of synthetic defects are specified as fractions of the area size. A shift of an element by 0.5 millimeters across an entire 8MP frame is undetectable, while within a 128×128 px area, it represents 5–6 pixels – a noticeable value.

Full Resolution

A 108–137 px area is transmitted to the network in a 128×128 format with minimal scaling. The entire frame would need to be compressed to 128 px, resulting in a loss of detail many times over.

The target boundary between "normal" and "defective"

Synthetic data for "normal" produces a variation of 0–2 pixels, while synthetic data for "defective" shows a shift of 5 pixels. The network learns to identify this specific boundary, rather than random variations in lighting.

Targeted Result

Each zone has a stable address (model, table, cell) and a name. A custom "rejector" rule can be assigned to any zone: an audible alert, a TCP message with the zone number, or a discrete camera output.

Alignment Possible in Any Orientation

The operator defines the alignment region – the area of the board with the components – on the image. Each frame is then aligned with the reference point, and the inspection areas are located on the new board, even if it has been placed at an angle or rotated by 180 degrees (section 4.4).

Defect originating from the board itself

A synthetic defect is not a drawn stain, but a displaced or rotated component along with a piece of the base. The model learns the relationship "component ↔ board," rather than the color of the stain. This is achieved in two steps (section 4.6).

Defects are reported gradually

Defect classes are only added where defects actually occur: solder bridges, missing components, misalignment, scratches. The model is retrained, while the grid remains the same.

1.2. Speed: The AI only looks at the relevant sections

The neural network never receives the entire frame. Only the rectangle corresponding to each zone is extracted, and the zone number is selected from two classes. Zones from a single frame are processed in parallel (up to four threads), and the results from all zones are displayed together as a single batch, rather than one zone at a time.

- **The analysis speed is determined by time.** The "Analyses per second (AI)" slider on the Camera tab indicates exactly N analyses per second, not "every Nth frame." The limit is the camera's frame rate and inference time.
- **GPU is used by default.** Inference is performed using ONNX Runtime with CUDA; if there is no NVIDIA graphics card, the program automatically switches to the CPU and indicates this in the status (`GPU (CUDA)` or `CPU`).
- **Clean buffers.** Memory allocated for clippings is used once per stream and then reused; analysis time remains consistent across frames.
- **Model for the task.** For ZONE-PAIR, the program automatically sets the input size to 128 px and uses the YOLO11 "s" size network: the extracted zone is barely scaled, and the network has sufficient capacity to handle hundreds of pairs of classes.
- **A live conveyor saves labor.** The operator doesn't need to manually move the parts through the process as frequently as in a traditional system; the parts are picked and aligned once, and then the cutting is performed.
- **The display does not interfere with analysis.** The tile is redrawn no more than 25 times per second, and only based on the new frame from the camera. The image is reduced to the size of the tile before being displayed on the screen, and the log is written to a file by a background thread. The interface remains responsive even with 8-megapixel frames and hundreds of zones.

Displayed above the video, in the status bar, are the actual figures: `fps` (frames per second) and the milliseconds of the last analysis, as well as `Eff`, which indicates how many analyses were actually completed per second.

1.3. What accuracy to expect: Let's calculate using our camera

The program does not guarantee a "0.5 mm" shift regardless of the camera and lens. The sensitivity to displacement is determined by how many pixels correspond to one millimeter of the object, and by the boundary that the model has been trained on. This is calculated in two lines:

```
px/mm = (frame width in pixels) / (field of view in mm)
Minimum perceptible shift ≈ 5 px ÷ (px/mm)
```

The number 5 px represents the lower limit of the synthetic shift in the recommended settings for anomalies (with a standard deviation of 0–2 px). The default settings are based on 11 px/mm: at this scale, 0.5 mm corresponds to 5–6 px, which is the target value that was agreed upon during development. Please enter your own values:

Camera (frame width)	Field of view 100 mm	200 mm	300 mm	400 mm
3264 px (USB 8 MP, as in the images)	0.03 mm/px → shift ≈ 0.15 mm	0.06 → ≈ 0.3 mm	0.09 → ≈ 0.45 mm	0.12 → ≈ 0.6 mm

2448 px (HIKROBOT 5 MP)	0.04 → ≈ 0.2 mm	0.08 → ≈ 0.4 mm	0.12 → ≈ 0.6 mm	0.16 → ≈ 0.8 mm
1920 px (SICK picoCam, HIKROBOT 2 MP)	0.05 → ≈ 0.25 mm	0.10 → ≈ 0.5 mm	0.16 → ≈ 0.8 mm	0.21 → ≈ 1.0 mm

Important. This is a bottom-up assessment of the element's displacement when there is a sudden change in the frame and stable lighting. To verify the actual boundary, use the "ZONE-PAIR: zones ↔ model" button and the recognition threshold (section 5.2): each zone has its own boundaries, and therefore its own level of certainty. The smaller the field of view and the more pixels the camera has, the smaller the defect the model will detect.

1.4. Quick Start: Settings pre-configured, zones and model ready in a few clicks

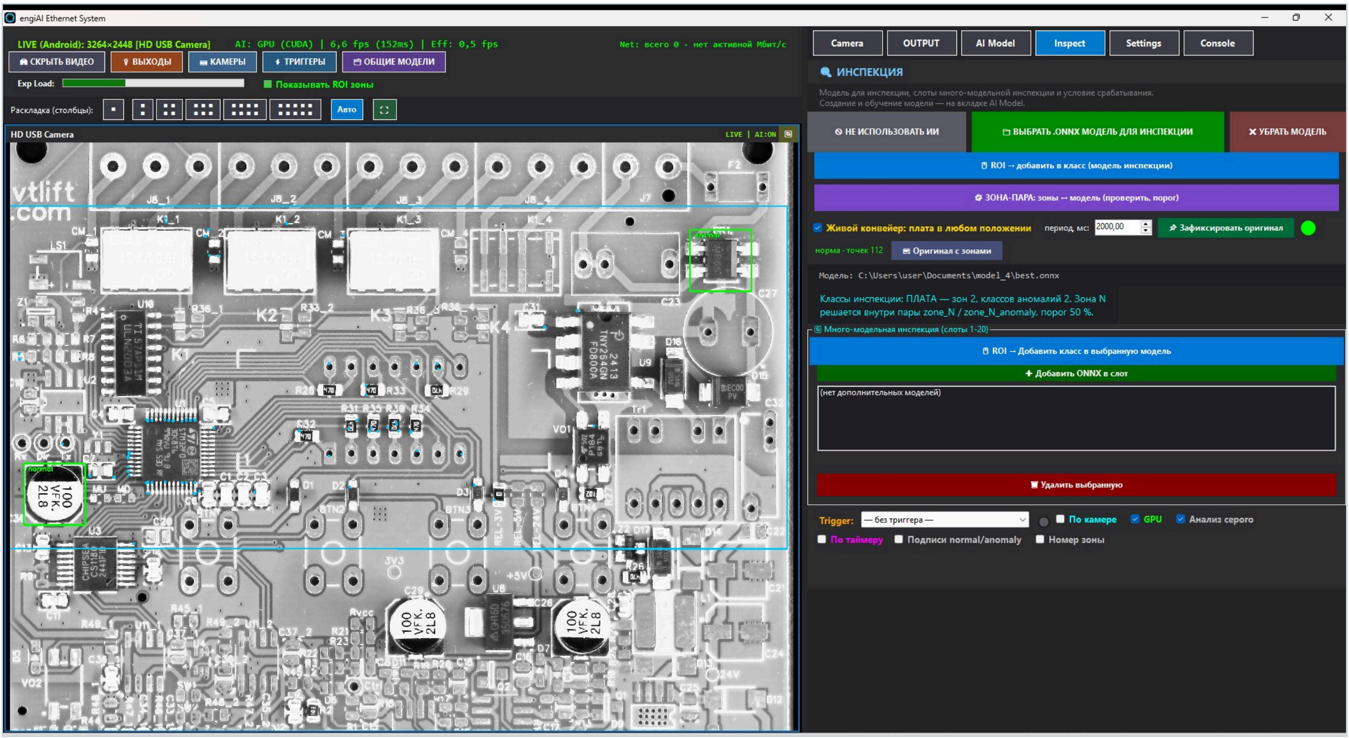
This method is designed to allow a technician with no prior experience in machine learning to obtain a working model in a single shift. Everything that could be pre-configured is already configured and stored in the program's database:

- **Zones per minute.** In the ROI editor, drag a frame around the element and specify the zone size (e.g., 256 px). If there are many elements arranged in a grid, drag the frame across the entire layout – the program will automatically calculate the rows and columns of the grid.
- **Synthetic data for normal and defective scenarios is pre-configured.** The "Default," "Element Offset," and "Color and Texture" buttons switch between pre-defined profiles. The number of files, camera frames, and expected time are displayed before the "Capture" button is pressed.
- **The training parameters are set automatically.** After the initial scan, the program sets the Image Size to 128, Model Size to s, Epochs to 50, and Batch to 64, and disables augmentations during training that conflict with the anomaly classes.
- **The camera is adjusted using a button.** The "Auto-setting for AI (per frame)" function automatically selects the shutter speed, amplification, gamma, and white balance; for USB cameras, there are "Auto-selection for forms" and "Set parameters" options.
- **Camera settings packs.** The selected settings are saved under a specific name and can be switched between from a list. The camera immediately accepts and stores these settings in its memory.
- **One ONNX file as output.** The trained model is automatically exported and immediately assigned for inspection; no further steps are required.

The entire process: connect the camera → create a model → mark zones → define the alignment region → take "Zone Images" on multiple functional boards → identify "Zone Anomalies" → TRAIN AI MODEL → Inspect tab. The remaining steps involve assigning sound, outputs, and TCP connections to the zones. To get started, you will need a computer with Windows 10/11 (a NVIDIA graphics card is recommended for faster inspection), a camera, one functional board, and one internet connection for installing the training environment. Defective boards are not required.

2. Program Interface in One Minute

The main window is divided into two parts. On the left is a live video feed from the cameras and a status bar above it. On the right is a panel with tabs, where all the controls are located. The user can drag the boundary between the two sections; the program remembers the size, position, and layout of all windows in its database.



Main window: on the left, a live view of the board with two zones (both green – "normal"), and on the right, the "Inspect" tab with the model model_4\best.onnx. The live conveyor is enabled, and the "normal · 112 points" indicator is displayed.

Buttons above the video

Button	What it opens
HIDE VIDEO	Removes the image from the screen. Analysis continues, only rendering is saved.
OUTPUTS	Reference window: contact information for discrete camera outputs from HIKROBOT and SICK, testing, connecting a load (section 7.1).
CAMERAS	"Camera Settings" window: adding, reconnecting, and deleting cameras, and viewing all parameters read from the hardware (section 3.2).
TRIGGERS	Connecting an external sensor to the camera's discrete input with a diagram and pinout (section 6).
GENERAL MODELS	Named sets "camera → model", assigned to the entire stand with a single selection (Section 8).
Layout (columns)	How many camera tiles to display in a row; "■" - single mode (selected camera zoomed in), "Auto" - based on the number of cameras, "⛶" - cameras across the entire monitor (Esc key).
Display ROI zones	Draw a grid of zones on the video.

Tabs on the right

Tab	Purpose
Camera	Parameters of the selected camera: shutter speed, amplification, color, resolution, automatic settings for AI, saved settings, analysis speed. The AI threshold is no longer available here - it is only on the "Inspect" tab.

OUTPUT	rejector: rules on zones - TCP packets to the robot/PLC, discrete camera outputs and audio alerts, event log, database.
AI Model	Creating a model: folder, classes, zone grid, zone snapshots, zone anomalies, reference boards, settings and training start.
Inspect	Working tab: which model is inspecting, inspection zones and threshold, live conveyor, which signal to capture, multi-model inspection slots.
Settings	Interface language (14 languages), license purchase, saving and loading settings, video recording.
Console	Event log: camera connection, training progress, rejector activation, warnings. The screen displays the 800-1000 most recent lines, and all data is written to the files logs\engiai-*.log.

Status bar

The status of the active camera is displayed above the buttons: `LIVE (Android): 3264×2448 [HD USB Camera] – type, resolution, and name; `AI: WAITING or `AI: GPU (CUDA) – the model's assessment and the number of milliseconds taken for the last analysis; `Net: total … active … Mbit/s – network load from GigE cameras (green up to 40% of the link, orange up to 70%, and red thereafter). The `Exp Load bar shows what proportion of the frame interval is occupied by the exposure time: if it reaches the right edge, the camera will not maintain the specified frame rate.

Each camera's tile displays a header with the camera's name, status (LIVE | AI:ON), and a reconnect button. Right-clicking on a frame opens the zone menu, while right-clicking and dragging allows you to pan across the zoomed-in frame.

3. Cameras: connection and configuration

3.1. Supported cameras

Type	How to connect	What can be configured from the program
HIKROBOT MV-CS (GigE, USB3)	Native MVS SDK. Opens strictly by serial number, the order of listing is irrelevant.	Exposure time: 6 µs, gain in dB, FPS, resolution (AOI), black level: 0...4095, all GenICam nodes (auto modes, gamma, sharpness, white balance R/G/B, digital shift, flip, GigE packets). Parameters are written in the UserSet1 of the camera itself. Discrete input Line 0, output Line 1, GPIO Line 2.
SICK picoCam (IDS uEye, GigE)	uEye driver. The camera should be visible in IDS Camera Manager with the status "configured correctly". The serial number is verified with the camera because the DeviceID in uEye changes with each start.	Gain, amplification 0...100, FPS, resolution, black level 0...255, pixel clock, gain boost, hardware amplification of channels R/G/B, hardware gamma, Bayer pattern. Parameters are written to the camera's memory. Trigger input on the optocoupler, Flash output (150 mA).
USB webcam (in the list - Android)	DirectShow via OpenCV, differ in moniker, so two identical cameras do not get confused.	Up to 8K resolution, FPS, all driver properties: brightness, contrast, saturation, hue, sharpness, gamma, amplification, shutter (auto), focus (auto), white balance (auto), backlighting compensation, etc. There are no discrete inputs and outputs; only TCP is available.
IP Camera RTSP	The link in the format `rtsp://admin:password@192.168.1.64:554/…` must be entered manually.	Shutter, amplification, resolution, and FPS are set in the camera's web interface. The program includes zones, model, and TCP.

There is no hard limit on the number of cameras: tiles can be arranged in a grid with any number; eight and nine cameras worked simultaneously on real setups. Each camera has its own model, its own zones, its own threshold, and its own analysis speed.

3.2. CAMERA WINDOW: Add, reconnect, delete

1. Click **CAMERAS** above the video. This will open the full-screen "SETTINGS FOR CONNECTED CAMERAS" window (to close, press Esc).
2. Click **+ Add Cameras**. The program will scan both the network and USB and display all found devices in a single list: `Hikrobot … SN: … IP:`, `SICK … SN: … DeviceID:`, `Android …`. SICK cameras take up to 4 seconds to be detected because the GigE device doesn't appear immediately in the driver. A SICK camera that is in use or incorrectly addressed will be marked with a ⚠ symbol and the reason.
3. Select the desired cameras and click "**Add**". To add an IP camera, click "**+ RTSP camera link**". While the cameras are connecting, a screen appears with a "Skip camera" button.
4. The camera will appear in the list with the line `name | type | serial number | LIVE or OFFLINE | AI:ON | Disp:ON`, and its video will be displayed in the left part of the main window.

The checkboxes below apply to the selected camera: **Camera enabled** (otherwise, it will be disconnected from the device and disappear from the screen), **Show video/zones** (this only controls the display; analysis always runs), **AI enabled for the selected camera**. The **Reconnect** button performs a complete disconnection and reconnection, restarting the AI stream and reloading the models.

The right-hand side of the window displays all the camera parameters read from the hardware upon connection, grouped and with units of measurement: "Show parameters for," **Re-read from camera**, **Copy to clipboard**. This is only for viewing; you can change the parameters on the "Camera" tab.

> **Important.** The HIKROBOT camera using the GigE protocol has a dedicated connection. After an unexpected program closure, it remains in use until the heartbeat timeout; the program waits for up to 25 seconds (10 attempts) and reports "camera is in use by another process." For SICK, if the IP address is incorrect, the program will attempt to correct it once, provided it is not within the adapter's subnet.

3.3. Camera Tab for HIKROBOT and SICK

The tab controls the camera, with the camera tile highlighted by a blue border. When the program is connected, it does not write any data to the camera; instead, the camera serves as the source of truth, and the sliders adjust to match its values. Settings can only be saved intentionally by pressing the "**Save settings**" button.

1. At the bottom of the tabs, enter the **Width (W)** and **Height (H)** of the frame (or select a preset from the list, which only contains the sizes supported by this camera), and click **APPLY RESOLUTION**. Typically, the full camera resolution is used. The size is rounded down to the nearest AOI step.
2. Click **Auto-Settings for AI (per frame)**. The program selects the exposure (priority) and amplification (no more than 40) so that the 99.5th percentile of brightness falls within the range of 220...248 without overexposure, then sets the gamma to 1.50, saturation to 115%, contrast to 105%, and performs auto white balance. Ticking the "auto" box enables continuous mode: checking and adjustment every 3 seconds, if the light changes.
3. If necessary, adjust the following manually: **Exposure** (slider in microseconds, label in ms, up to 1 second), **Gain**, **FPS**, **Black Level**. For SICK, additionally: **Pixel clock** (determines the maximum FPS and minimum exposure time), **Gain boost**, **Hardware gamma**, **WB Red / Green / Blue**.
4. Click **Save Parameters**: the file `camera_params\<SN>.ini` and the entry in the camera's memory – UserSet1 for HIKROBOT (with the default settings), and EEPROM for SICK. The result is displayed on the button itself ("✓ Saved"). For HIKROBOT, there is a checkbox **Automatic saving to the camera (UserSet1) after 5 seconds of change**.

"Hikrobot: Camera Parameters (GenICam)" block

Only the data captured by the camera is displayed. The following settings are available: Auto Exposure (Off / Once / Continuous), Auto Gain, Gamma and Gamma Mode, Sharpness, Hue, Saturation, Auto White Balance, White Balance by Channel, Digital Shift, Black Level Enabled, FPS Limit Enabled, Horizontal and Vertical Flip, GigE: Packet Size, GigE: Inter-Packet Delay, GigE: Bandwidth Limit, Test Image. The "**Read Parameters from Camera**" button updates the table.

"IMAGE COLOR" block

Bayer pattern (Auto / BGGR / RGGB / GRBG / GBRG), Saturation 0...200 %, Gamma 0.30...3.00, Contrast 50...200 %, reset button. Software processing is applied to all frame consumers: display, recording, inference, and training images, so the image is the same during training and inspection. For HIKROBOT, the pattern is taken from the camera pixel format, while for color SICK sensors, it is taken from the sensor and refined based on the first frame.

Parameter	What it does	What it affects
Analysis per second (AI)	Upper limit of inspection speed: 1...200	Lower values result in lower strain on the graphics card; independent of camera connection.
AI threshold:	Not set on the "Camera" tab.	Zone recognition threshold: only on the "Inspect" tab (section 5.2), one per camera.
Exposure	Exposure time:	Shorter exposure times result in less blurring of moving objects; a value of zero is not allowed.
Gain / Gain boost	Matrix signal amplification:	Increases brightness, but also adds noise.

Pixel clock (SICK)	Sensor clock frequency:	Changes the FPS and exposure ranges, which are re-read.
FPS	Frame rate in free mode	Not used in trigger mode
Width / Height	Frame size (AOI)	Smaller frame = higher frequency and less traffic

> **Important.** Take pictures for training in the same conditions as the line will operate: the same lighting, the same speed, the same camera position. After changing the settings, the program reminds you in the console that the model needs to be retrained on the new images.

3.4. Camera Tab for USB/Webcam

For USB cameras, the tab displays the "USB Camera: Resolution, FPS, and Properties" block. Properties of the driver that the camera does not provide are not displayed; the field boundaries are set by the program because DirectShow does not report ranges.

1. **Resolution** and **Apply**. The **Check** button cycles through standard resolutions from VGA to 8K, and only displays the resolutions that the camera accepts. The "Camera Format" line shows the actual resolution and FPS.
2. First, allow the auto-exposure, auto-focus, and auto-white balance settings to adjust, then press "**Set Parameters (Disable Auto, Save as Is)**": this will disable auto mode, and the program will automatically select the appropriate manual exposure setting to match the current brightness. The values and format are saved in the database and will be applied automatically each time this camera is connected.
3. Series **With AI**: **Automatic selection based on shapes** (20–60 seconds, adjustment of exposure and contrast based on live frames, criterion – clear outlines without overexposure), **Shapes, without color**, **Shapes, soft**, **Return**, **Typical**.
4. The values entered in the fields are immediately sent to the camera, without requiring the user to press Enter, and are then immediately written to the database. The "**Read properties from camera**" button indicates that the camera has successfully received the data.

> **Suggestion.** For AI, it's better to remove color not by adjusting the camera's saturation, but by using the general option **Train in grayscale (overall model setting)** on the Camera tab (it's also called "Train in grayscale" on the AI Model tab). This will make both the training images and the inspection images grayscale, while the camera will continue to provide color information for the operator. The model in the example was trained in grayscale.

3.5. Saved Camera Settings (Sets)

The "Saved Camera Settings" block is displayed for USB, SICK, and HIKROBOT cameras. By selecting the parameters for the product, save them under a name: for another board or different lighting, simply select a set from the list.

- **Save as...** – Reads the current values from the camera and prompts for a set name. If the name is already in use, it offers to overwrite it.
- **Selecting a set from the list** immediately writes the parameters to the camera and saves them: for USB, to the database; for SICK and HIKROBOT, to a parameter file and the camera's memory (UserSet1 / EEPROM). Then, the panel reads the values from the hardware and displays how many parameters have been accepted and how many have been deviated from.
- **Overwrite** overwrites the selected set with the current values, **Rename** and **Delete** only affect the database entry; the camera settings remain unchanged.

The kit includes: for USB – resolution, FPS, driver properties, and auto-modes; for SICK – gain, exposure, FPS, black level, pixel clock, gain boost, white balance R/G/B, hardware gamma, and software color; for HIKROBOT – gain, exposure, FPS, black level, and all GenICam nodes. The frame size of industrial cameras is not included in the kit; changing it restarts the capture. The kits are the same for all cameras of the same type, so one kit is placed on each camera on the stand.

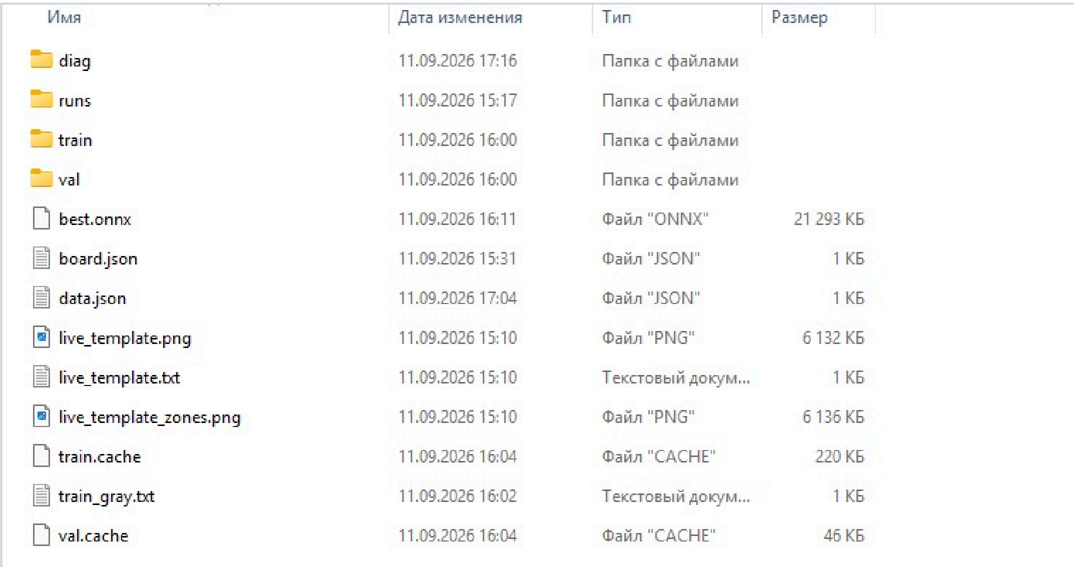
3.6. GigE Network and Reconnection

- The HIKROBOT GigE packet size is automatically set to optimal when opened; it can also be manually adjusted in the GenICam block. Jumbo frames (8164–9000) reduce the load; when using fragmented frames, set 1500 and increase the inter-packet delay.
- The `Net` line calculates the load from each camera based on its own data (useful load × frequency) and compares it to the actual incoming network traffic. A significant discrepancy indicates dropped frames or extraneous traffic. Virtual adapters and NDIS filters (GigEVision, uEye, Npcap) are excluded from the calculation.
- If the camera is visible but the stream isn't working, click "**Reconnect**" (in the CAMERA window or on the tile). The most common causes are network bandwidth limitations or USB bandwidth issues.
- The SICK picoCam loses its settings when the descriptor is closed; if there is no configuration file in the camera itself, the program applies the configuration file and prompts the user to click "Save Settings" to save them to the camera.

4. ZONE-PAIR Model: From Grid to Trained Model

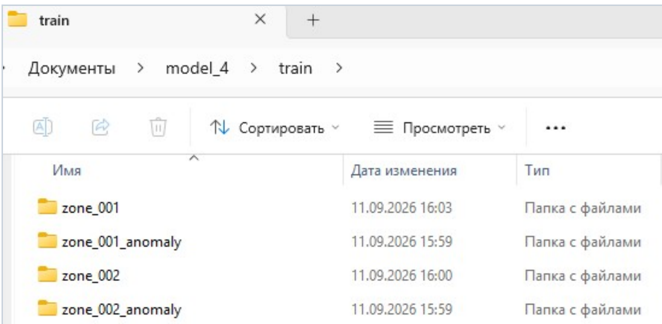
4.1. How the method works: One model, one class per zone

The ZONE-PAIR model is a standard YOLO11 classification network, with a number of classes equal to the number of zones, plus optional anomaly classes. A class consists of a folder within the `train` and `val` directories. The program automatically creates all folders; no manual creation is required. Here is an example of a model's folder structure after training:

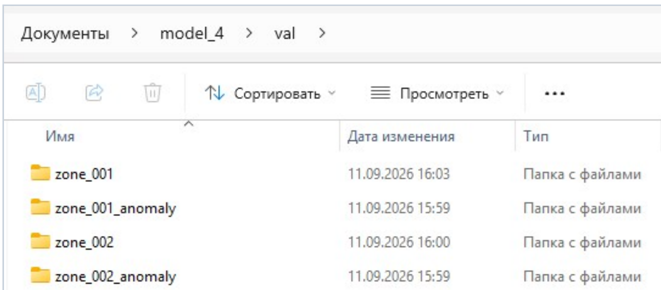


Folder model_4: train and val – images, data.json – regions, live_template – reference boards, board.json – threshold, best.onnx – trained network.

```
model_4
  Training\                        Training images
    zone_001\                   Normal zone 1 (C30): real_… from the camera, syn_… synthetic data
    zone_001_anomaly\           Defect zone 1: syn_…_combo1_d, _rotate1 …
    zone_002\                   Normal zone 2 (BR1)
    zone_002_anomaly\           Defect zone 2
  Validation\                          Validation set: the same four folders, automatically populated
  data.json                     Zones: tables, coordinates, size, color, enabled/disabled; classes; slots
  live_template.png / .txt      Reference boards: frame, alignment region, red (excluded) zones
  live_template_zones.png       Reference with drawn zones and numbers
  board.json                    Zone recognition threshold ({ "threshold": 0.50 })
  train_gray.txt                Grayscale "set" feature
  best.onnx                     Trained model - assigned for inspection
  runs\  diag\  *.cache         Auxiliary: training progress, images for analysis, training cache
```



train: Two folders per zone – normal and anomaly.



val: The same structure, the program automatically fills it during capture.

The files named `real_…jpg` are actual images captured by the camera (and are included in both the training and validation sets). The files named `syn_…jpg` represent synthetic data. The files named `edit_…jpg` are images taken from the zone editor. In the example provided, after capturing images of several boards: `zone_001` contains 502 files in the training set and 111 in the validation set; `zone_001_anomaly` contains 680 files in the training set and 144 in the validation set; `zone_002` contains 813 files in the training set and 181 in the validation set; `zone_002_anomaly` contains 340 files in the training set and 60 in the validation set.

> **One storage location.** All model markings – inspection zones, alignment regions, classes, and slots – are located only in the model folder: `data.json` and `live_template.txt`. There are no more zones in the program's database, so the ROI editor on the AI Model and Inspect tabs displays the same zones, and the model folder can be copied to another computer without losing anything. Old models are automatically transferred to the file upon the first opening.

The program automatically recognizes the model as a "payment" based on the presence of the `zone_NNN` class. The class number is the zone number within the grid (1...N). Therefore, the main rule of the method is:

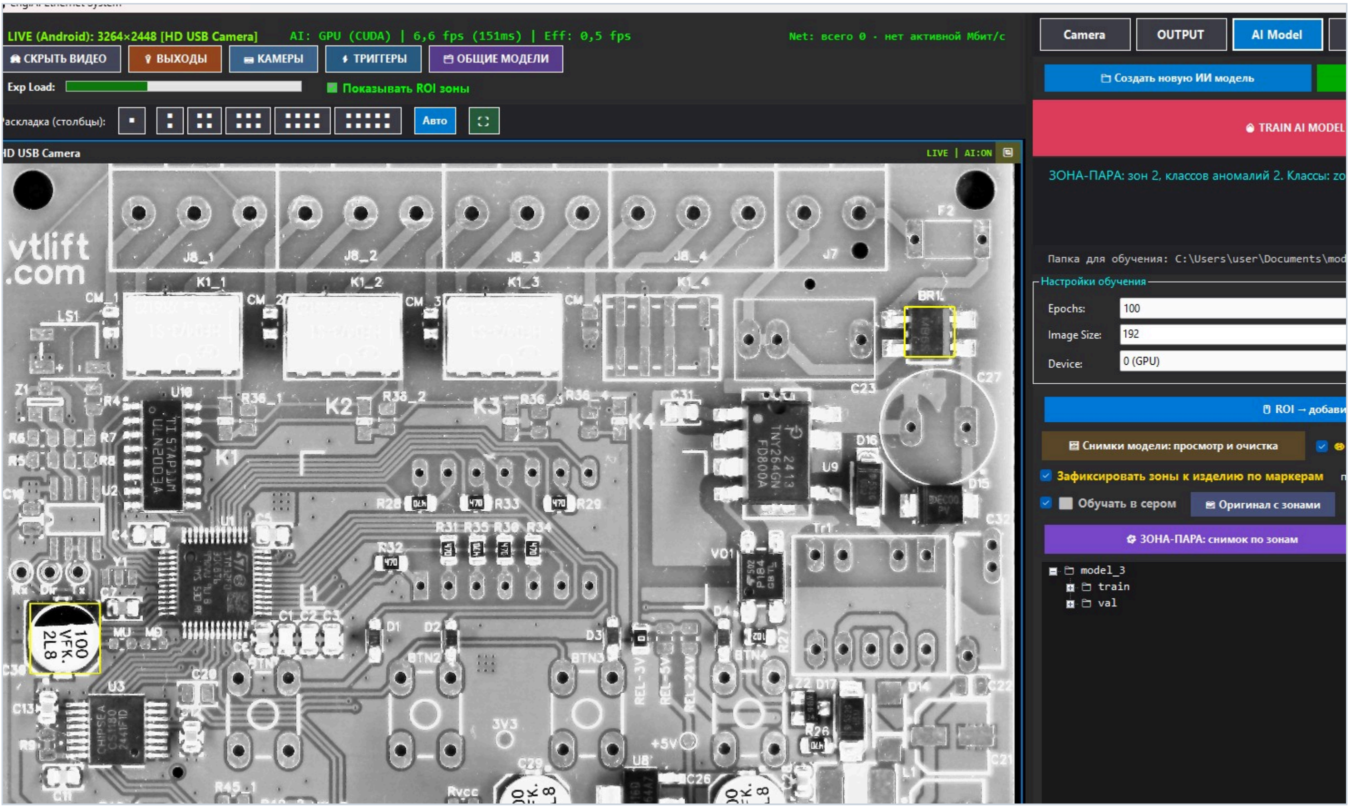
> **Important.** Zones are part of the model. Once the classes have been created, they cannot be modified (added, deleted, or rearranged): the zone numbers will no longer match the class numbers, and these zones will receive the tag `NO_ZONE_CLASS`. If the zones need to be changed, create a new model.

4.2. Create the model

1. Click on the "**AI Model**" tab and click "**Create New AI Model**".
2. Specify the parent folder and model name (the program suggests using a free `model_N`). The default list of classes contains `normal` and `anomaly`; for ZONE-PAIR, these are not needed – the zone classes will be created automatically when the first zone-based snapshot is taken. You can delete them or leave them empty: before training, the program will suggest removing empty classes.
3. Click **Create Model**. Two folders, `train` and `val`, will appear on the disk. The model will open for editing (it will not be loaded for inspection). If a model already exists, instead, click **Select Model Folder**.

4.3. Mark zones on the board

1. Place the correct board directly under the camera, in the exact position it will occupy on the line. Then, click **ROI → Add to Class (Training Model).**
2. In the "ROI to Class" dialogue, select any class (this is irrelevant for labeling) or enter a new name and click "**OK + Open ROI**". Later, all the zone classes will be visible here, grouped by the template `zone_###`: clicking on a group selects it, and a second click deselects it; the "**Delete Group**" button removes the entire group along with the images.
3. The "ROI Selection" window will open. Drag a box around the area you want to measure using the left mouse button. A field labeled "**Area, px**" will appear; enter the size (which will default to a multiple of 32, for example, 256), and press Enter – the box will become a square with the specified size. Repeat for each area you need to check. In the example, there are two areas: around C30 and around BR1.
4. If there are many details and they are arranged neatly, a grid is more convenient: stretch the frame across the entire product and enter the dimensions of the area – the assistant will show you how many rows and columns you have. Alternatively, in the **Zones Grid** block, specify the number of rows, columns, width, and height of the cells, and click **Create Grid**. You can drag the grid precisely onto the product using the mouse.
5. The right panel displays each zone table, showing its size and a "enabled" checkbox. An inactive zone does not generate a class, and a completely hidden table is not involved in the inspection. The buttons **All zones ✓ / ✗**, **Color**, **Delete**, **Delete ALL zones**. Closing the window with the Enter key will save the zones in `data.json` and display them on the video in yellow.



Result: The marked zones appear in the AI Model tab: two yellow zones on the live image – BR1 on the top right and C30 on the bottom left. To the right are the ZONE-PAIR model buttons, the "Fix zones to product with markers" checkbox, and the training settings.

> **Advice.** The component should occupy a significant portion of the area, with a small amount of board remaining: traces, labels, pads. This "base" is then used in synthetic data (section 4.6). For 0603–1206 components at 11 px/mm, a size of 128 px works well; for capacitors, bridges, and connectors, a size of 192–256 px, as in the example, is recommended. The network input size (Image Size) should then be approximately equal to the area size.

Red zones: Exclude a zone from synthetic data with a single click

On the "**AI Model**" tab, click on the already created area directly on the video with the left mouse button: the area will turn **red** and will no longer be part of the zone-based and anomaly detection. A second click will restore it. The file counters in the synthetic data windows are recalculated immediately, and the "off N" label above the frame indicates how many areas are turned off. The list of red areas is stored per camera and model; on the "Inspect" tab, the same area remains normal (with its own yellow areas – section 5.5).

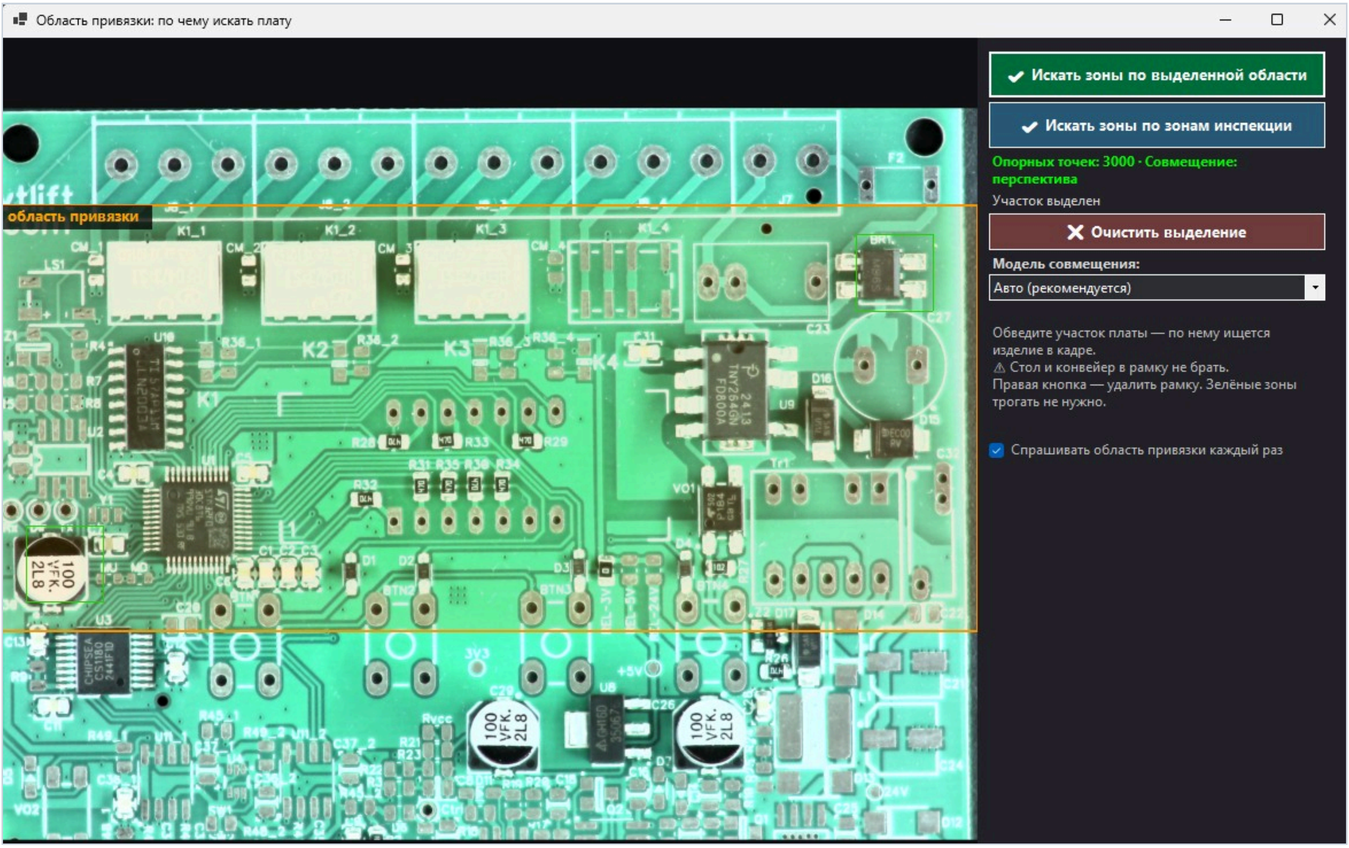
4.4. Fix reference: alignment region

The inspection zones are defined based on the position of a single circuit board. To allow for the insertion of another board with a shift and rotation, the software remembers a **reference** point: a live frame and an **alignment region** – a section of the board with features that are used to locate the product in each new frame. Once the area is found, the software transfers the inspection zones to the new board. This is how the image acquisition, synthetic data generation, and inspection processes work.

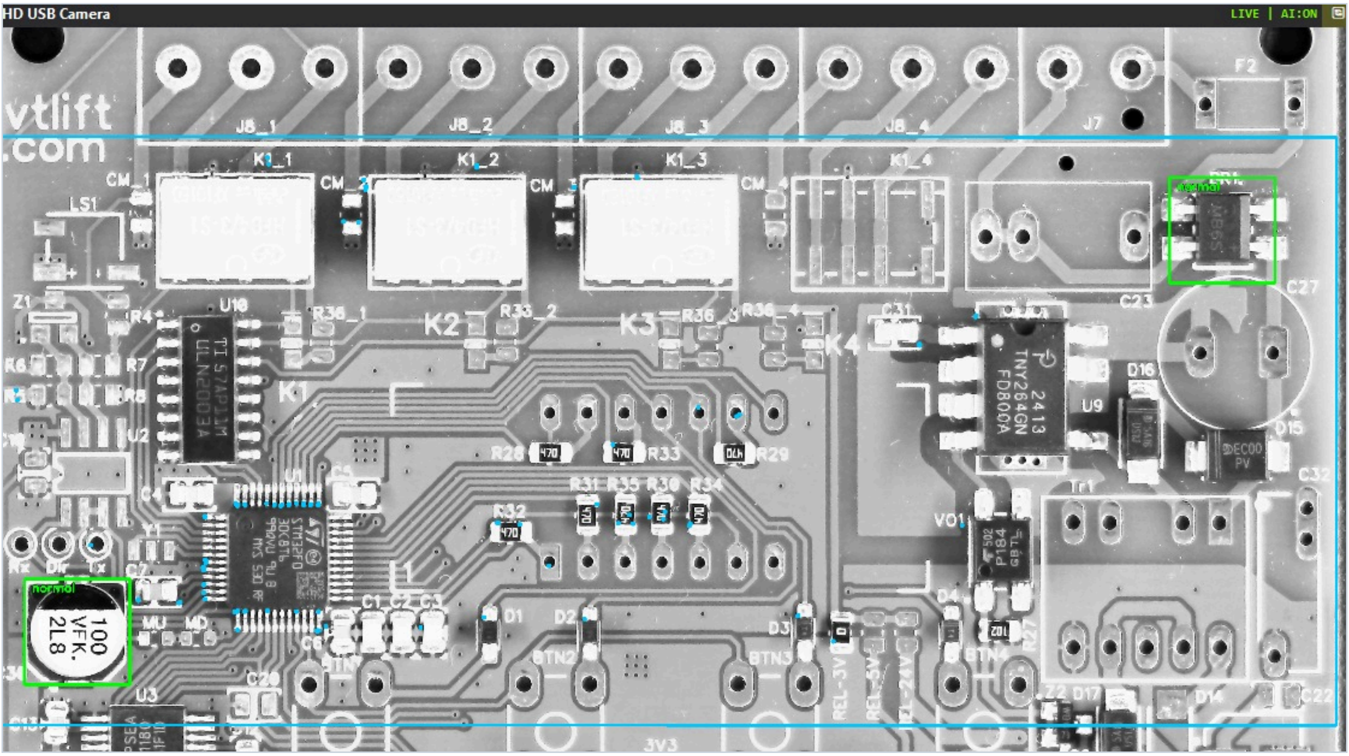
> **Two different rectangles. Inspection zones** represent what the model checks (green areas). **Alignment region** is the marker used to locate the board (orange frame). Reference points should never be placed within the inspection zones: if the board were to be located by searching for the capacitor, a shifted capacitor would pull the inspection zone along with it, making the defect invisible. Therefore, in the alignment region, use silkscreen, holes, traces, and nearby microchips, but not the table and conveyor belt.

1. On the "AI Model" tab, check the box "**Associate zones with the product using markers**" and specify the "**search period, ms**" (default: 100). Place the board under the camera and click "**Capture reference**".

2. A window titled "Alignment region: where to look for the board" will open, displaying a live image and green inspection zones. Drag the frame across the board area with the details using the left mouse button. You will immediately see "Reference points: N · Alignment: rigid / perspective": there should be hundreds of points, and if "too few – expand the frame". The new frame replaces the old one, and the right mouse button removes it.
3. Click **✓ Search within selected area**. The second button **✓ Search within inspection zones** selects the area based on the zones themselves, only when there is nothing else nearby. If the checkbox **Ask for alignment region each time** is unchecked, the next fix will repeat the previous area without the window.
4. The program writes "Reference established. Zone N, 1 anchor points, points K" and saves `live_template.png` and `live_template.txt` (frame, area, red zones). The checkbox **Show shooting zones on frame** displays the identified zones on the video, while the anchor points used for alignment are shown as blue dots – this allows you to see which points the program is "anchored" to.



Alignment region: orange frame – area of the board with components (3000 reference points), green – C30 and BR1 inspection zones, which are not within the frame.



Live view after alignment: zones are found on their components and labeled "normal", blue lines – alignment region boundaries, blue points – matching reference points on the board (which do not fall within the zone).

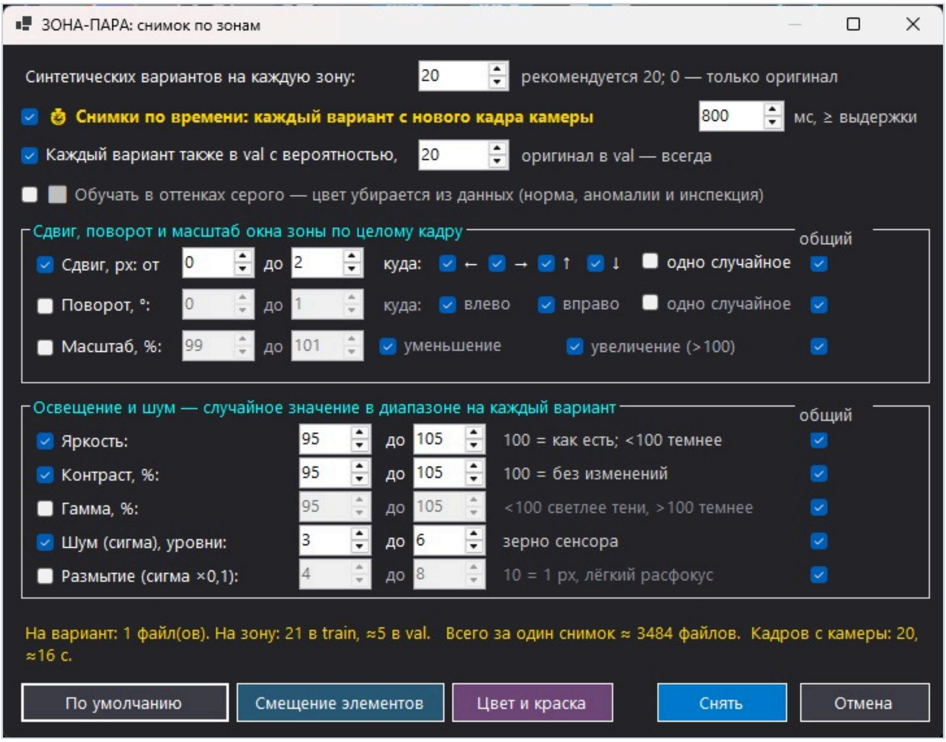
Alignment model	When to use
Automatic (recommended)	The board is aligned rigidly (translation, rotation, scale); the perspective is only connected if it aligns significantly better. Accuracy in a small alignment region – 1–2 px.
Rigid	The camera looks straight down, the board lies flat. The most stable option if the alignment region is small.
Perspective	The camera is at an angle or the board is tilted. A large area with points across the entire width is needed, otherwise the homography will "skew".

The "**reference with zones**" button opens the saved reference with a grid and zone numbers (section 4.8). The found component is checked for reasonableness: the area should be between 0.5 and 2 times the area of the reference, it should be a convex quadrilateral, and the support points should occupy at least a quarter of the area along each axis – otherwise, the result is discarded, and the message "component not found" is displayed in the console.

> **Important.** Previously, areas where the "zones" were located were defined by the number of zones on the screen – this was because the payment was being tracked by individual zones. However, when re-establishing the reference with the alignment region based on the details of the circuit board, the number of identified normal zones on the board has increased from 46-51 to 106-112 out of a total of 134.

4.5. Zone-based image: normal and its synthetic data

The "**ZONE-PAIR**" button: creating a zone-based image involves cutting the live image into zones, placing each zone's excerpt into its corresponding class `zone_NNN`, and adding synthetic data representing normal conditions. Synthetic data representing normal conditions is not a defect; it reflects variations in the image, such as a slightly different placement of the board, slightly different lighting, and sensor noise.

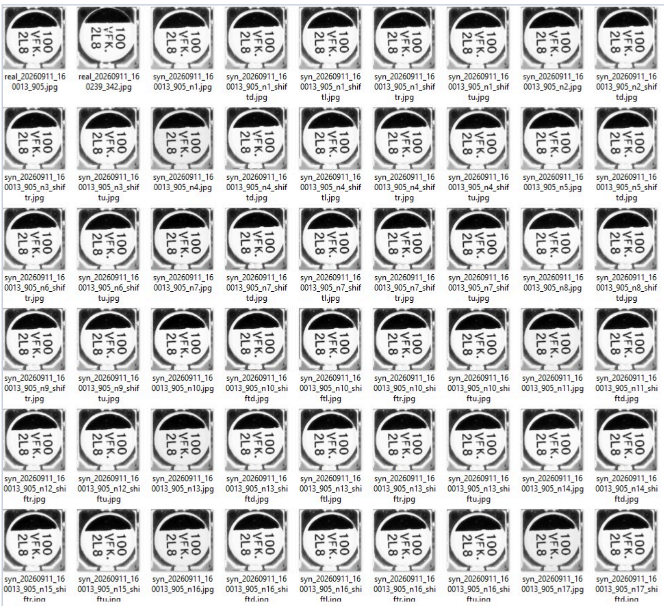


"ZONE-PAIR: Zone-based image selection": Number of options, images by time, percentage in val, window zone shift/rotation/scale relative to the entire image, with "destination" selection, lighting and noise, and a final file counter. The window is expanded using the mouse.

Settings	What to create and default settings
Synthetic options for each area	How many synthetic files to create per cutout. Recommended: 20, 0 (only reference), maximum 500.
Time-stamped images: each option from a new frame from the camera	Enabled, period 800–1200 ms (at least the exposure time). Each option is taken from a new live frame: real sensor noise, vibration, and light flicker instead of being drawn.
Each option also in val with a probability of %	20% reference always appears in val: Ultralytics requires each class to be in both folders.
Train in grayscale	Overall model flag: color is removed from normal, anomalies, and inspection. It is not possible to mix color and grayscale files in one model; the program will warn you.
Shift, px · to where	The sliding area shifts across the entire frame by 0…2 px in the specified directions (← → ↑ ↓); the missing strip is taken from the adjacent area on the circuit board, and the cutout remains even. The checkbox all directions provides a separate file for each direction, while one random provides a single file.
Rotation, ° · to where	By default, it is disabled: no rotation is applied to the alignment region, and enabling it blurs the boundary with the defect. The options "left/right" and "all directions/random" function similarly to the shift option.

Scale, % · decrease/increase	By default, it is turned off. Two checkmarks represent two sides: reduction (<100) and increase (>100, the zone window takes neighboring frames). Each marked side corresponds to a separate file.
Brightness/Contrast/Gamma	Random value from the range for each option: brightness 88...112, contrast 88...112 %, gamma 98...102 % (in the image); in the "Element Offset" profile, the brightness is wider to avoid being mistaken for a defect.
Noise (sigma) / Blur	Sensor grain 3...8 levels, slight defocus 0.4...0.8 px.
"General" column	With the checkbox, the view is included in the general file along with the others; without it, a separate "reference + only this view" file is created. The more individual views, the more files are created per option.

1. Click **ZONE-PAIR: image by zones**. The line at the bottom of the dialog box pre-calculates: "For option: 1 file. For zone: 21 in train, ≈5 in val. Total ≈ 52 files from the camera: 20, ≈16" (for two zones).
2. Select a profile: **Default** (offset 0–2 px, brightness ±5, contrast 95–105 %), **Element Offset** (offset 0–1 px, brightness −10...10) or **Color and Tone** (offset 0–2 px, brightness ±2, contrast and gamma disabled to preserve the hue as a key feature).
3. Click "**Remove**". The generation process runs in the background with a progress window showing "Zone 1/2 · 26 files · Frame 4/20" and a cancel button. Upon completion: "Done: 2 zones → classes zone_001...zone_002, files: 52". The program immediately sets the training settings to the specified values: Image Size 128, Model Size s, Epochs 50, Batch 64 (section 4.9).
4. Repeat the image capture using the same button on several different, undamaged PCBs. For a reliable model, you need at least 80 files per class, and most importantly, several actual examples (5-10 PCBs), rather than a single image with hundreds of variations.



train\zone_001 - a standard C30 font in the conductor's compartment: real_... with camera settings syn_..._n1...n20, including shift operations (shiftd, shiftl, shiftr, shiftu). The text is always positioned correctly relative to the base.

train zone_002 — standard BR1: real_... with three platforms and options for slight turning (rotl, rotr) and shifting.

4.6. Synthetic data defect: element shifts with the base

The synthetic data is based on observation. The model learns particularly well when, instead of a drawn-out stain, it is presented with **the inspected element itself, along with the surrounding components of the circuit board:** including silk screen lines, tracks, color, and the texture of the base. The program reduces the frame to the size of the element and then **shifts or rotates this area**, filling the edge with adjacent pixels from the frame. As a result, the element is not in its original position relative to the base: exactly as the inspector would see it in a real defect, with real sensor noise, real lighting, and a real circuit board.

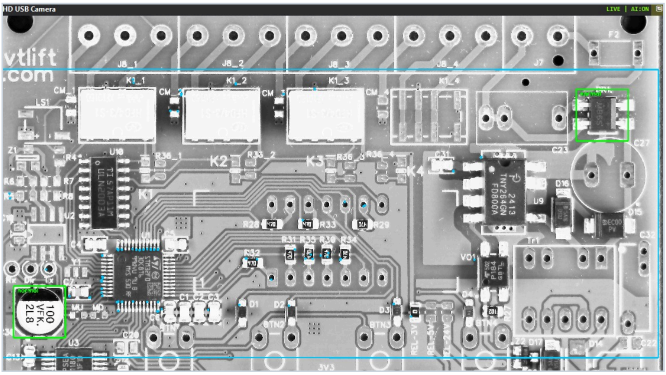
The "normal" and "anomaly" zones within a region differ **only in the position of the element relative to the base.** Everything else is the same, so the network learns the relationship "element ↔ board," rather than the shade of a spot or the lighting.

Two clicks:

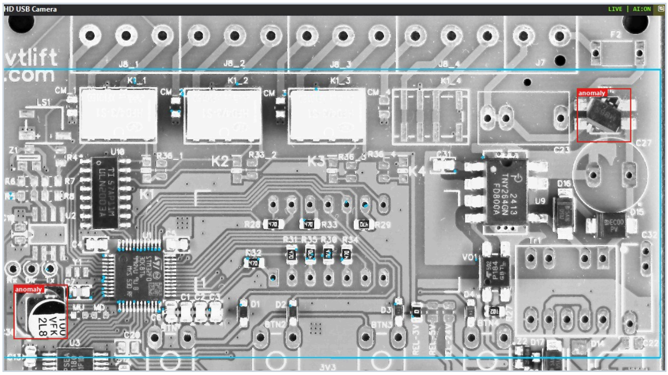
1. ZONE-PAIR: image by zones → Capture. Normal extracts from each zone, with 20 different capture variations (window shift 0–2 px, brightness, noise), are assigned to the `zone_NNN` classes.
2. ZONE-PAIR: Zone anomalies → Create. The default settings already include both types of offset: **zone shift** of 5–12 px in all directions and **rotation of the element area** (70% of the zone, circle) by 15–180°. The files are placed in the `zone_NNN_anomaly` classes.

No further configuration is required: you don't need to draw defects or search for faulty boards. Next, TRAIN AI MODEL.

Before and after: what the program achieves

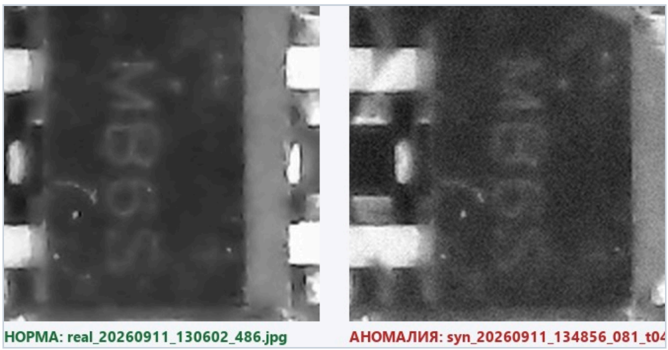


BEFORE: a live image of a good PCB. Both areas are green – "normal".



AFTER: the same areas are populated with synthetic data from the anomaly folders – C30 is shifted to the right and down, BR1 is rotated. Both areas are red – "anomaly". The PCB will also look like this in the case of a real defect on the line.

Large, zone 1 (C30): normal from the camera on the left, synthetic data on the right – the part and base are shifted down and to the right, the edge is filled with neighboring elements from the image.

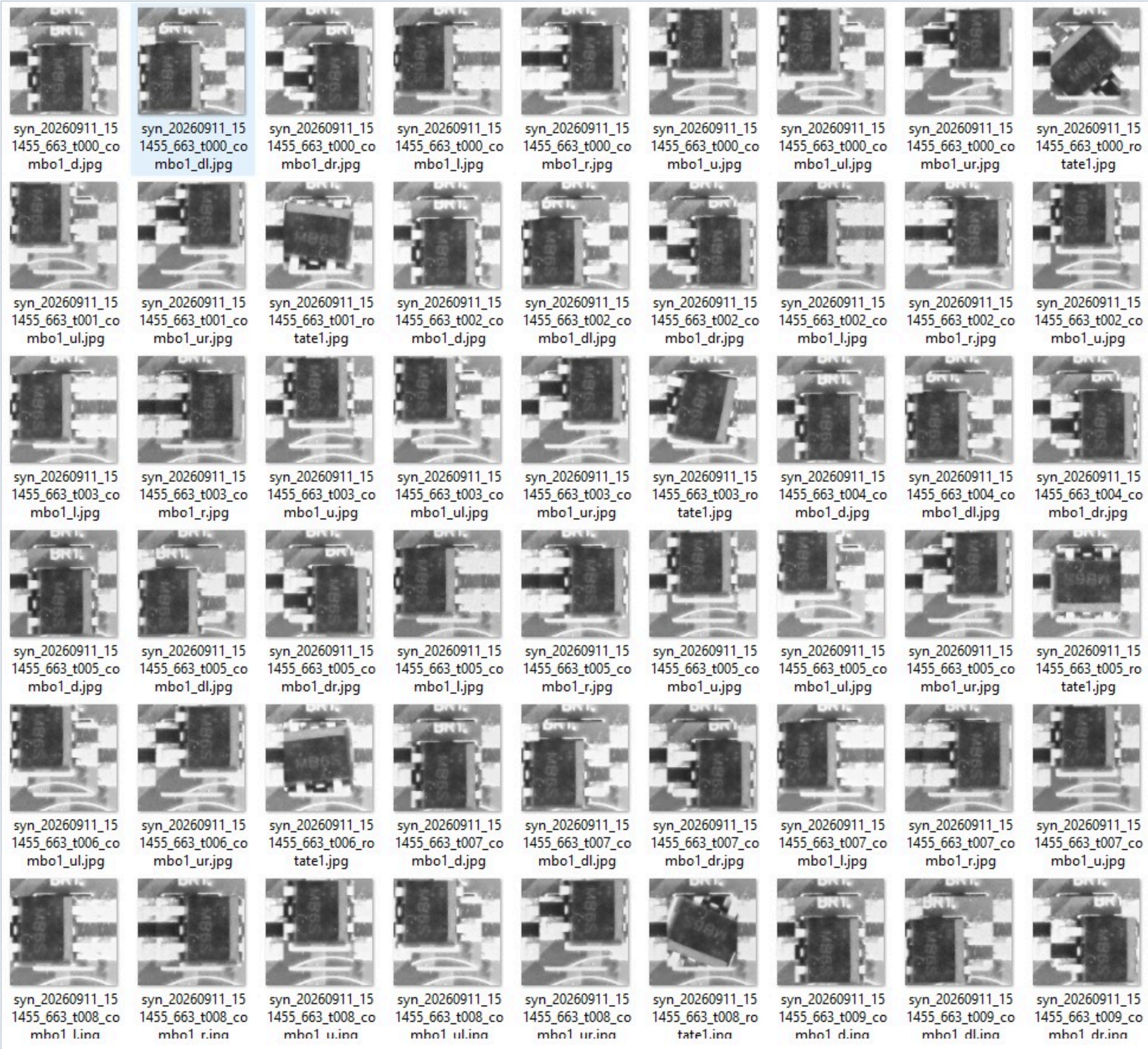


Large, zone 2 (BR1): the area of the part is rotated 180° – the pads are in the correct place, the body is facing the wrong way.

Folders representing anomalies as seen by the user

Compare with the "normal" folders from section 4.5: the only difference is the position of the detail relative to the base.

train\zone_001_anomaly — defect C30. In each copy of t000, t001... there are nine files: combo1_d, _dl, _dr, _l, _r, _u, _ul, _ur – the area has been shifted by 5–12 px in eight directions (d = down, u = up, l = left, r = right), rotate1 – the capacitor area has been rotated.



Train zone_002_anomaly – Defect BR1: The chassis has detached from the base and is present on each of the eight sides, rotated at a random angle. The PCB and pads are genuine.

> **How to read file names.** `syn_<date>_t003_combo1_dr.jpg` – Copy 3, general file, down-right shift (d = down, r = right, u = up, l = left); …`_rotate1.jpg` – Element area rotation; …`_scale1_dn/_up` – Scale in the larger or smaller direction; …`_tl/_c/_br` – Where the area was located (corner, center). The name always shows which specific defect is in the file, and the extra data is removed using the "Delete synthetic data (train + val) class" button in the zone window.

Detail marking: why is the "thick" line and how is it configured?

The parts have markings – labels and 1–2 px thick lines within the frame. These markings vary from batch to batch: different fonts, labels on capacitors are rotated, and some symbols are erased. If a synthetic defect, such as a "scratch," is drawn with the same thin line, the system will not be able to distinguish it from the marking line and sensor noise, and will begin to reject boards from a different batch.

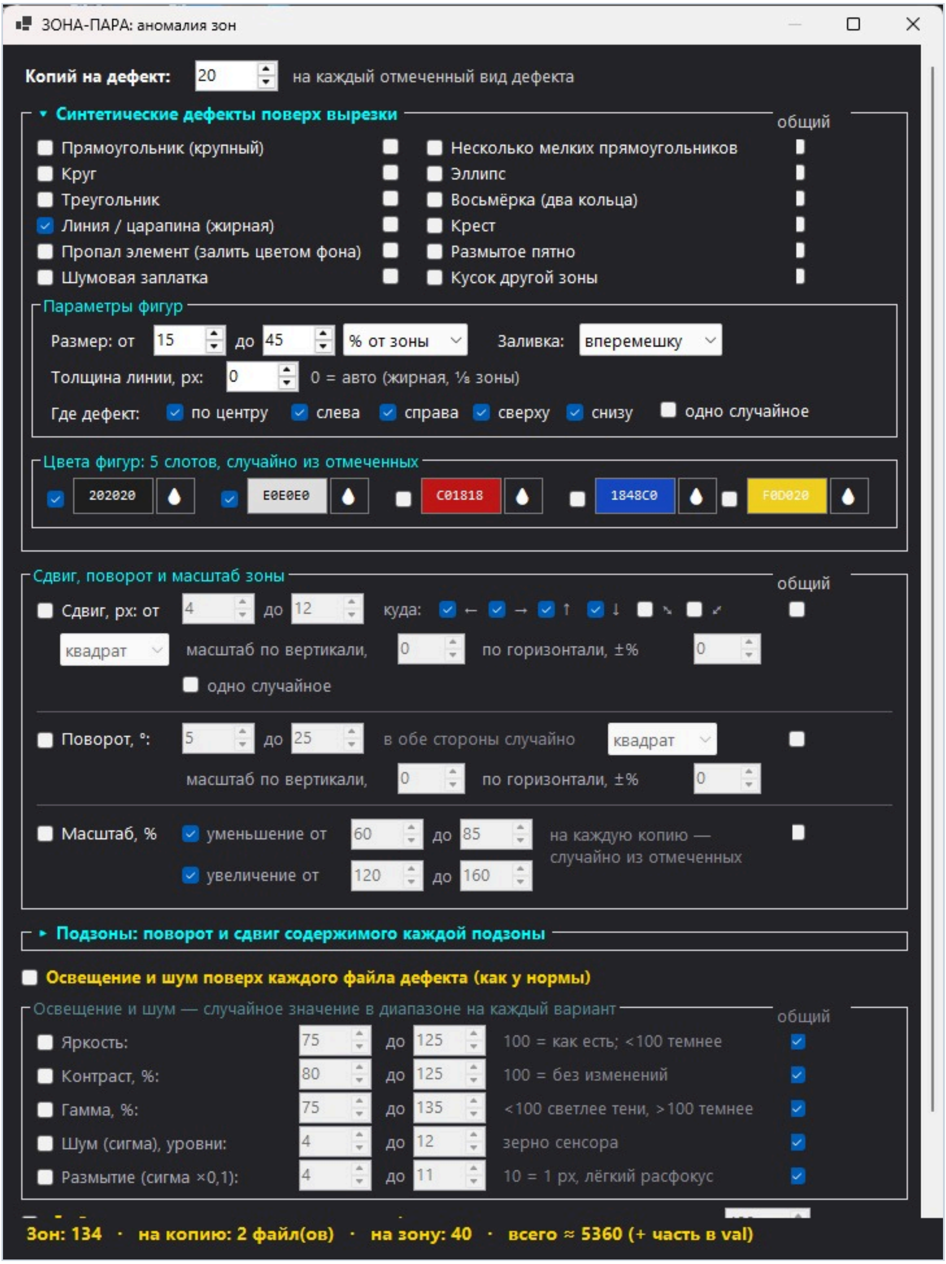
The synthetic data ZONE-PAIR addresses this from both sides. First, the main defect – the displacement of the element – moves the marking along with the body, so the marking does not become a distinguishing feature: it looks the same in both normal and anomalous states. Second, the default **line/scratch** is thick (⅛ of the zone, no less than 3 px) and is clearly not a marking stroke. The thickness is now explicitly defined: the **Line thickness, px** (0 = auto) field applies to both the line and the cross.

Hand-drawn defects (rectangles, circles, lines, "missing element," blurred spots, and others) are grouped together in a collapsible section **Synthetic defects on top of the cutout**. By default, it is collapsed and disabled: the main defect on the board is misalignment. Enable it for specific tasks: flooded area, extra drop, worn-off marking.

Figure parameters	What it defines
Size: from ... to, % of the zone / px	The size or length of the figure. In percentages, it is proportional to the smaller side of the zone; in px, it is the same for all zones, which is convenient when the actual defect size in pixels of the frame is known.
Fill	solid / outline / mixed (outline is also thick).
Line thickness, px	Line / scratch and cross. 0 — auto (thick). For marking details, do not go below 3 px.
Where the defect is	Center, left, right, top, bottom – where to place the piece (with a slight random variation). Checkmark **all provisions** - a file for each marked location; **A random occurrence** - One file is selected randomly from the marked files. The file counter at the bottom of the window automatically reflects this selection.
Figure colors: 5 slots	Five slots instead of a list of names. A checkmark indicates that the slot is active, and the color is randomly selected from the marked options (±12 per channel, to avoid a single shade). The "sample" button allows you to choose a color from the palette, while the "dropper" (represented by 💧) allows you to click on a live camera frame (the wheel controls zoom, and the right button controls panning) to select a color from a real object or component. Two slots are enabled by default: one for dark and one for light colors, while the other three are disabled.
"General" column	The marked defect type is placed in the general file along with other "common" types (e.g., a scratch on an element that has been shifted). Without the checkbox, the file will contain "normal + only this defect". The "all positions" view will always be in a separate file.

4.7. Zone Anomalies: Settings Window

The `**ZONE-PAIR**` button: "Anomaly Zones" creates classes `zone_NNN_anomaly` from normal cuts. All zones are selected, except for the red (disabled) zones (created by clicking on the AI Model frame in section 4.3). Anomaly classes are not needed everywhere: add them to zones where defects actually occur, and the model will check the remaining zones for recognition.



The "ZONE-PAIR: zone anomaly" window: copies the defect on top; a collapsible group of drawn figures with parameters, location, and palette; shifting, rotating, and scaling the zone with the shape and scale of the area ±%; subzones; lighting over the defect; time and val snapshots. The window expands with the mouse.

Settings	What it does
Copies the defect	One number for all types: how many copies with new random parameters to create for each marked type (default: 10).
‣ Synthetic defects on the cutout	Default group: rectangle, small rectangles, circle, ellipse, triangle, octagon, line/scratch, cross, `**missing element**`, blurred spot, noise patch, piece from another area. Figure parameters, location, and palette – Section 4.6.
Shift, px · where · shape	Main defect of the board. The area (circle or square) shifts by 5–12 px in the specified directions: ← → ↑ ↓ and the two diagonals ↘ ↙. scale vertically / horizontally, ±% – the size of the area relative to the zone: 0 – the entire zone, -30 – an area 30% smaller (only the element shifts, the base remains), +30 – an area larger than the zone (captures neighboring elements from the frame). all directions – for each side, according to the file.
Rotation, ° · shape	The area (circle or square, same scale ±%) rotates to a random angle in both directions.

Scale, % · decrease/increase	The area as a defect: reduction of 60–85% or increase of 120–160%. Each marked side is its own file. Usually disabled on the board.
"General" for shift and rotation	If both checkboxes are selected, it means **one element in one file**: the translation and rotation are applied through a single transformation (not sequentially, otherwise it would result in a "mess"), the area scale and shape are common and defined by the translation.
▸ **Sub-zones**	The area is divided into 2, 4, or 9 parts, and within each, a separate area rotates or shifts; the "separate photo for each sub-zone" option creates a file for each part. This is needed for large areas with multiple elements.
Lighting and noise on each defect file	Enabled and set to normal: the defect should differ in geometry, not in lighting.
Photos over time / in val	Each copy from the new camera frame (800 ms), 20% in val, the first file in the val class is always present.

1. Click **ZONE-PAIR: zone anomalies**. The result is predetermined: "Zone: 2 · per copy: 2 files · per zone: 20 · total ≈ 40 (+ part in val)". With "all directions" enabled, the file shift per copy becomes nine, as in the folders above.
2. Select a profile: **Default** (with a zone shift of 5–12 px + rotation of the area, figures, and subzones are collapsed and disabled), **Element Shift** (only a shift of 4–12 px, with normal brightness), or **Color and Paint** (figures, "missing element", colors from the palette, without any shift). The window immediately remembers any changes made, with each click.
3. Click **Create**. The classes `zone_NNN_anomaly` will appear in the model tree. To add a defect from the line to the same classes, use ROI → Add to class.

> **Important.** The lower limit of the synthetic shift (5 px) is immediately above the normal range (0–2 px). Do not expand the normal range to 5 px or narrow the anomaly to 2 px: the boundary will disappear, and the model will fluctuate on real boards. If you need different sensitivity, change both limits consistently and convert them to millimeters (section 1.3).

4.8. Reference with zones: solder pads and zone editor

The "**reference with zones**" button opens a saved reference image with a grid. Synthetic data is created here without a camera: the zoom function, the right button for panorama, and the left button for selecting zones (Ctrl to add, drag with the mouse to define the area, Ctrl+A to select all), and double-clicking opens the zone editor.

- **Solder pads for selected areas...** – "Solder pads: short circuit between tracks". The program automatically identifies tracks within the area (Ozu threshold, light-colored strips ≥ 2 px), placing a solder pad in the gap between adjacent tracks, using the color of the tracks, with a dark border and a highlight. Settings: between vertical or horizontal tracks, pads for images 1...3, size 8...18% of the area, images per area (5), where to place – in an anomaly or in normal, surface lighting, in val – a new image with different lighting or an exact copy.
- **Zone Editor**: Tools: Brush, Pipette, Color Spot, Copy Area, Paste, Stamp, Area for pads; palette: "solder / pad", "board", "shadow / mask", white, black; "Copy" field and **Create Image** button. This allows you to draw a defect that was not present on the line: a filled pad, an extra element, or a missing label.

The editor's files are named `edit_`... and are categorized into either the "zone" class or its "anomaly" class. On the "Inspect" tab, the same window opens, but it is only for viewing purposes.

4.9. Model training

The "**Training Settings**" section on the AI Model tab is automatically populated after the initial scan based on the specified zones: Image Size 128, Model Size s, Epochs 50, Batch 64. These values correspond to the 108–137 px zones. For the 256 px zones, as in the example, set Image Size to 192 or 256 – this will minimize compression (the Epochs 100 and Image Size 192 settings are shown in the main window). Verify the values and click "**TRAIN AI MODEL**".

Parameter	What it means	How to choose

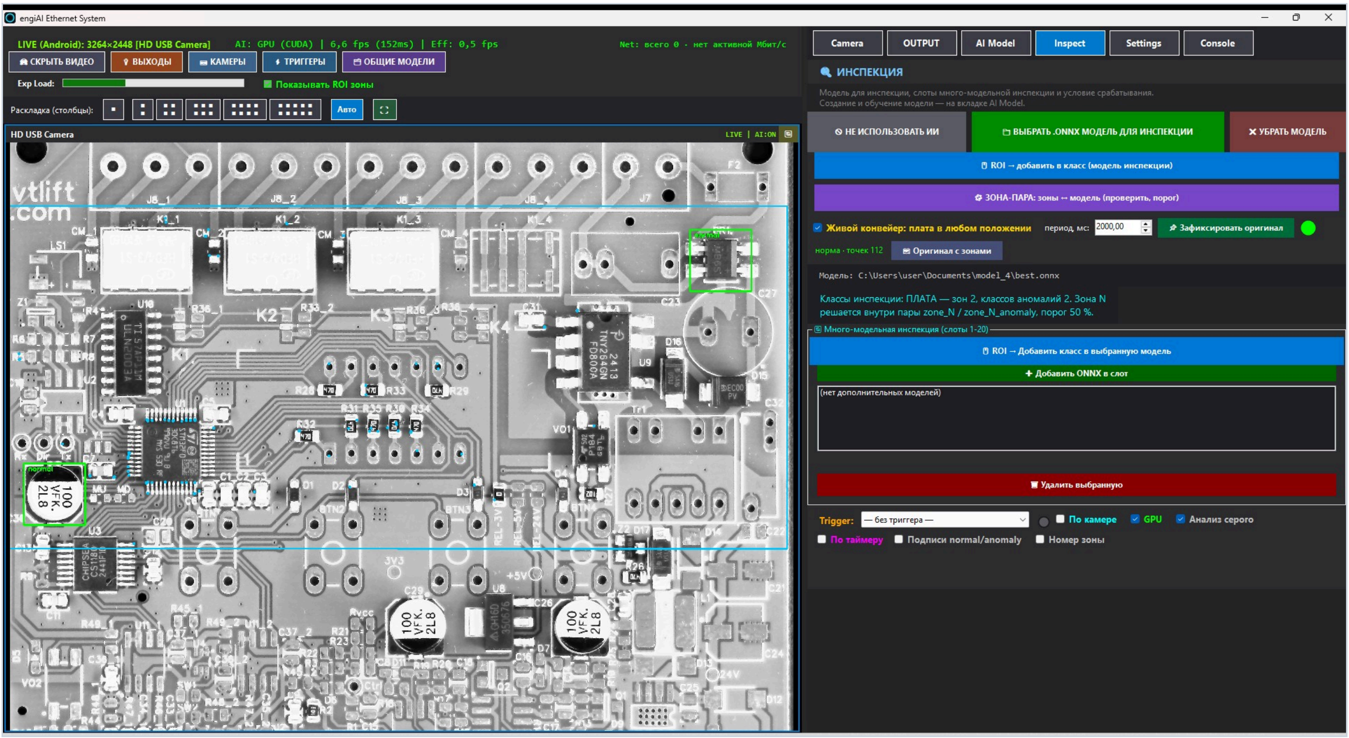
Epochs	How many times the model will process the entire dataset	50 for the default board; 100-150 if there are many anomaly classes
Batch Size	How many images per step	64; reduce if there is insufficient video memory
Image Size	What size are the crops, divisible by 32	Approximate zone size: 128 for a 108-137 px zone; 192-256 for 256 px zones, as in the example
Model Size	n / s / m / l / x	s — optimal for hundreds of pairs of classes; n is insufficient capacity, m and above — slower
Device	On what to train	0 (GPU) with an NVIDIA graphics card; without CUDA, the program will automatically switch to the CPU
Workers	Number of data preparation threads	12 (default for a multi-core processor)

1. Before starting, the program checks the following: it suggests deleting empty classes, unnecessary folders in val, and identifies areas in val that are empty. When the program is first launched, it sets up the training environment (Python 3, ultralytics, onnx, onnxslim, onnxruntime packages) – this requires an internet connection and a few minutes.
2. For the ZONE-PAIR model, training does not require erasing, flipping, cropping, or randaugment: these augmentations conflict with the anomaly classes (an inverted zone is already a different zone). The corresponding line is written to the console.
3. The button turns red when **STOP AI MODEL** is pressed, and the ring on the right shows the percentage per epoch. The training progress can be seen in the Console. For two zones and 100 epochs on a graphics card, this takes approximately 10–15 minutes.
4. After completion, the program exports `best.onnx` to the root of the model folder and immediately assigns it for inspection. The "**ONNX Export**" button repeats the export without retraining.

> **Advice.** After adding a defect or new data, the retraining process is initiated using the same button: the dataset is augmented, and then the AI model is retrained best.onnx It is being overwritten. Please do not touch the grid during this process.

5. Inspection

All the work is done on the "**Inspect**" tab. The model has already been trained; now it's necessary to check the regions, set the threshold, select the speed source, and, if the board arrives in any orientation, enable the "live conveyor" feature.



In the "Inspect" tab: model model_4/best.onnx, line "Inspection classes: BOARD - zones 2, anomaly classes 2. Zone N is resolved within the pair zone_N / zone_N_anomaly, threshold 50%", live conveyor with a period of 2000 ms and "normal" indicator with 112 points, below: speed source and "GPU", "Gray analysis" checkboxes. In the frame, both zones are green - "normal".

5.1. Assign the model and check the zones

1. If a model is not assigned, click the "**SELECT ONNX MODEL FOR INSPECTION**" button and specify `best.onnx` the model folder. The "Inspection Classes" field will display all the classes; the names are taken directly from the ONNX file, so the output numbers are guaranteed to match the names.
2. Click on **ZONE-PAIR: zones ↔ model (check threshold)**. The window displays: "Number of zones in the grid: 2 / Number of classes zone_NNN in the model: 2, number of anomaly classes: 2 / Each zone in the grid has its own class ✓". Warnings such as "Zones without a class in the model" or "Classes without a zone in the grid" indicate that the grid has been modified after training.
3. **DO NOT USE AI** temporarily disables analysis on this camera, without deleting settings (each camera has its own flag); **REMOVE MODEL** removes the model and slots completely.

5.2. Zone recognition threshold

For each zone, the network provides probabilities for all classes, but the decision is made only within each pair. The probability of the "`normal`" class, q, is calculated as p(zone_N) / (p(zone_N) + p(zone_N_anomaly)) and is then multiplied by a "confidence" factor, which represents how likely the pair is to be considered a valid output by the network (full confidence is achieved at 10% or higher). The resulting score, ranging from 0 to 100%, is compared to a threshold: a score above the threshold indicates a "normal" state, while a score below the threshold indicates an `anomaly`.

The default threshold is 50%, and it is stored in `board.json` alongside the model and applied immediately without requiring a restart. Below 50%, the threshold shifts towards normal more strongly: 25% → a cutoff of 6%, 10% → 0.4%, 1% → almost all green; 99% only allows through confident areas. This scale is needed so that the operator can distinguish between 5% and 99%, rather than seeing only areas in red.

> **Advice.** Set a threshold on a set of 10 known-good boards: increase it until all zones remain green, then test on boards with known defects. The program writes a line to the console every 10 seconds: "[ZONE-PAIR] ... normal N, anomaly M, threshold 0.50; worst q: z17:0.31 ..." – this shows which zones are closest to the threshold.

5.3. Live pipeline: payment in any position

Halo **Live Conveyor: Finding the PCB in Any Position** includes searching for the PCB based on its alignment region before each analysis: the frame is aligned with the reference image, the cutting zones are extracted from the aligned image, and the alignment region and zones (green – normal, red – anomaly) are drawn on the video. The field **Period, ms** specifies how often to search for the PCB (default: 1000, 2000 in the image). The indicator next to it indicates: "normal · N points", "ANOMALY · N points", or "PCB not found" or "No reference – confirm". The verdict is based on a single analysis, without accumulation.

The reference and alignment region are fixed using the same button **Fix reference**, as when taking a picture (section 4.4); **reference with zones** opens it for viewing. The boundaries of the alignment region and the blue reference points, which align the board, are drawn on the video. Areas that extend beyond the frame after alignment are marked with the `SKIP` label.

5.4. Analyze based on which signal

The source of the speed is always the same: when a new one is included, the program removes the previous one.

Method	How to enable	When to use
By camera	"By camera" checkbox	Continuous monitoring; speed limited by the "Analyzes per second" slider on the Camera tab
By timer	"By timer" checkbox and interval (ms)	Periodic monitoring, for example, once per second
By trigger	Select a connection from the "Trigger" list and a front ("analyze + on input" or "analyze -")	Camera-based conveyor: a frame is captured at the moment the product arrives – the most accurate method (Section 6)
GPU	Checkbox: "GPU"	Enabled by default; toggling reloads the model
Grayscale Analysis	Checkbox: "Grayscale Analysis"	Enabled automatically for models trained in grayscale; masks are converted to grayscale before inference

5.5. Yellow Zones: Turn off zone by clicking

In some cases, certain zones may need to be temporarily removed from control during the process. This might be because a particular element is not yet installed in this batch, a zone is blocked by a fixture, or the system is undergoing debugging. To do this, you don't need to modify the grid or model; instead, the zones are simply disabled directly in the video feed, specifically within the "**Inspect**" tab.

1. Click on the area on the camera tile with the left mouse button: the area will turn **yellow** and will no longer be analyzed. A second click will re-enable it.
2. Group of zones: Hold down the left button and drag the frame across the board. All zones within the frame are turned off; if they were already yellow, the frame turns them back on. This also works when zooming with the scroll wheel.
3. The list of yellow zones is stored in the database by camera and model, and persists after program restarts and camera reconnection. The console displays the line "[Zones] ... turned off zones: N".

The actions of the yellow zones are determined by the main checkbox: "**Yellow (muted by clicking) zones are silent**" in the "Audio" tab of the OUTPUT window (enabled by default):

Checkbox:	Behavior of Yellow Zones

Enabled (default)	The zone is not analyzed at all: the mask is not sent to the network, the OFF label is displayed, there are no sounds, no TCP packets, and the camera does not exit. In a live pipeline, a yellow zone does not affect the "normal / anomaly" verdict.
Off	The yellow border remains only as a marker: the area is being analyzed and functions normally. This is convenient for marking areas to be checked without disabling the monitoring system.

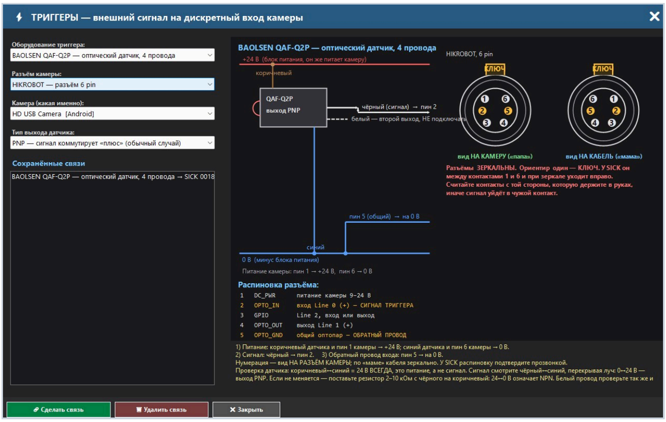
> **Advice.** It is not necessary to delete the rejector rules in the yellow zones: when the checkbox is enabled, they simply do not process any data, and after being disabled, the zone will immediately start working.

5.6. What you can see on the screen and in the console

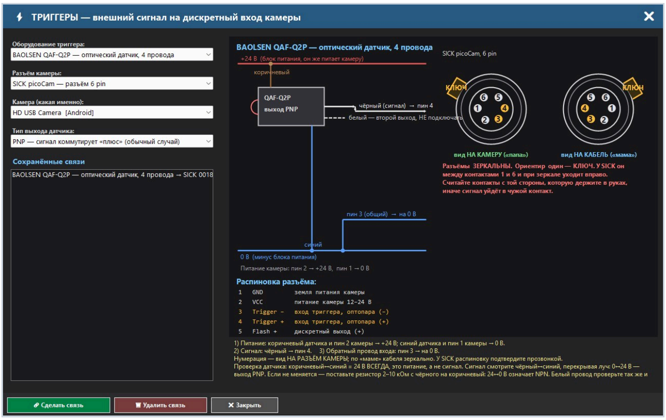
- The result for each zone is displayed; the border is colored according to the zone's status (red for active, green for normal). When there are more than 50 zones, only the normal status is displayed in green, while anomalies and uncertain zones are indicated with labels. Hundreds of labels on each frame made it difficult to read the image and increased the rendering load. Yellow zones are disabled by clicking (section 5.5).
- The "Trigger Analysis" panel at the bottom of the tab displays each trigger event as a row: `` `[hh:mm:ss.mmm]`` Door Camera + TRIGGER 4ms | AI 12ms #57`, along with the result for each zone. These rows are not written to the database.
- Right-clicking on a region in the video: "Classify this region as '...' → Rename the model's response for this region (this does not modify the model itself, but the rejector and log will see the new name); Also, "Create a rejector for this region" and the rule counter.
- Check the "**Save inspection images**" box. The folder: images with marked areas are saved as files with the extension `inspect_….jpg`. The settings for this feature vary for each camera.
- Every 10 minutes, a frame from the live conveyor and a text list of zones are written to a subfolder named `` `diag` ``. The last 20 images are also stored there, for reviewing disputed cases without expanding the folder.

6. Triggers: External sensor on the camera's input

Main mode for the conveyor: the camera captures a frame based on an electrical signal from the sensor's discrete input. This is the most precise moment of capture: the product is always in the same position in the frame, and the analysis is performed only once per product. The **TRIGGER WINDOW** collects the "sensor → camera" connection and draws a wiring diagram for the selected connector.



TRIGGER Windows for HIKROBOT (6 pin): PNP sensor, signal to pin 2, common pin 5, pinout.



The same applies to the SICK picoCam (6-pin): signal to pin 4 (Trigger +), common pin 3 (Trigger −).

1. Click **TRIGGER** above the video. Select **Trigger Device** (e.g., BAOLSEN QAF-Q2P optical sensor, 4 wires; any PNP or NPN sensor can be connected in the same way), **Camera Connector** (HIKROBOT 6 pin or SICK picoCam 6 pin), **Camera (specify the model)**, and **Sensor Output Type**: PNP – switches the "positive" signal (standard case), NPN – switches the "negative" signal, a 1-4.7kΩ resistor is required on +24V.
2. Assemble the circuit according to the diagram: power the sensor and camera from a single 24V power supply, connect the sensor's signal wire to the camera's input, and connect the input's return wire to 0V. The white wire of the sensor is the second output, which should not be connected.
3. Click "**Create Link**". The link is saved in the database; duplicate links are not created. The "**Delete Link**" button removes the selected link.
4. On the "**Inspect**" tab, select the trigger from the "**Trigger**" list and the front: "analysis + on input" (frame when 24V appears) or "analysis - on input". The program switches the camera to trigger mode: for HIKROBOT, TriggerMode = On, TriggerSource = Line0, while SICK stops the capture, sets the mode, and restarts. The circle next to it indicates the input level.

Identifying and labeling the connectors

Pin	HIKROBOT MV-CS, 6 pin	SICK picoCam, 6 pin
1	DC_PWR – Camera power supply: 9–24 V	GND — camera power supply ground
2	**OPTO_IN — Line 0 input (+), trigger signal**	VCC — camera power supply (12–24V)
3	GPIO — Line 2, input or output	**Trigger input – optocoupler (–)**
4	OPTO_OUT — Output of Line 1 (+)	**Trigger + – Trigger input, phototransistor (+)**
5	**OPTO_GND — a common optical fiber, with a return path**	Flash + — discrete output (+)
6	GND — camera power supply ground	Flash – — discrete output (-)

> **Important.** The "male" connectors on the camera and the "female" connector on the cable are mirrored. The only reference point is the key (a large opening in the connector); for SICK, it is located between contacts 1 and 6, and when mirrored, it goes to the right. Identify the contacts from the side you are holding. An incorrect connection means a burned-out input: the power and output are located nearby. Verify the SICK wiring diagram by testing the connections.

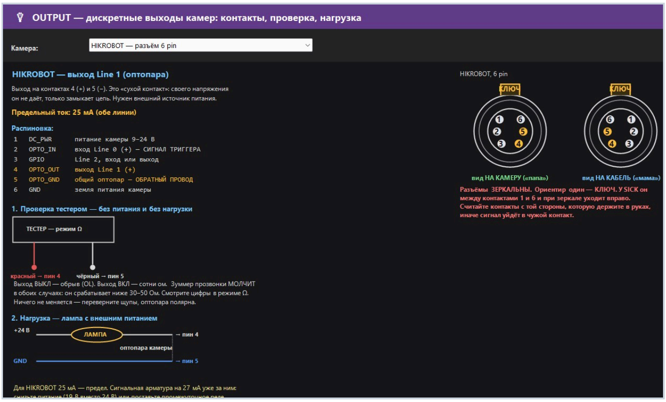
> **Advice.** When measuring with the "+", measure the peak, not the trough: the variation in the peak-to-trough ratio on a real device was 4.2 ms, while the peak-to-peak ratio was 40.7 ms. Sensor check: the brown and blue wires should always be 24V, which is the power supply; the signal should be between the black and blue wires, with the beam crossing. If it doesn't change, place a 2-10kΩ resistor between the black and brown wires; 24V↔0V indicates an NPN transistor.

7. Outputs: rejector, camera discrete outputs, and TCP

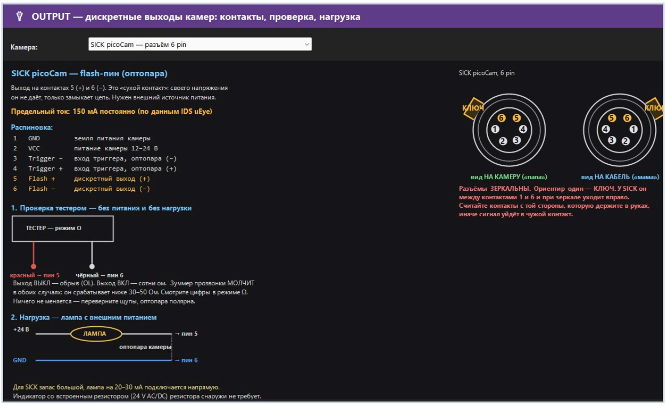
The rejector converts the inspection result into an action. The rule is created for a specific area (or for all areas of the model, class, or marked areas) and is triggered when the model detects the specified class in that area. The actions are three: sending a TCP packet to the robot or PLC, switching the camera's digital output, and issuing an audible alert to the operator. Each trigger is recorded in the log.

7.1. OUTPUT REFERENCE: Contacts and testing with a multimeter

The "**EXIT**" button on the video opens a help window that does not configure anything: it shows which contacts are used for the output, how to test it without power, and how to connect a load.



HIKROBOT — Line 1 output (photo-relay) on pins 4 (+) and 5 (−), maximum current 25 mA.

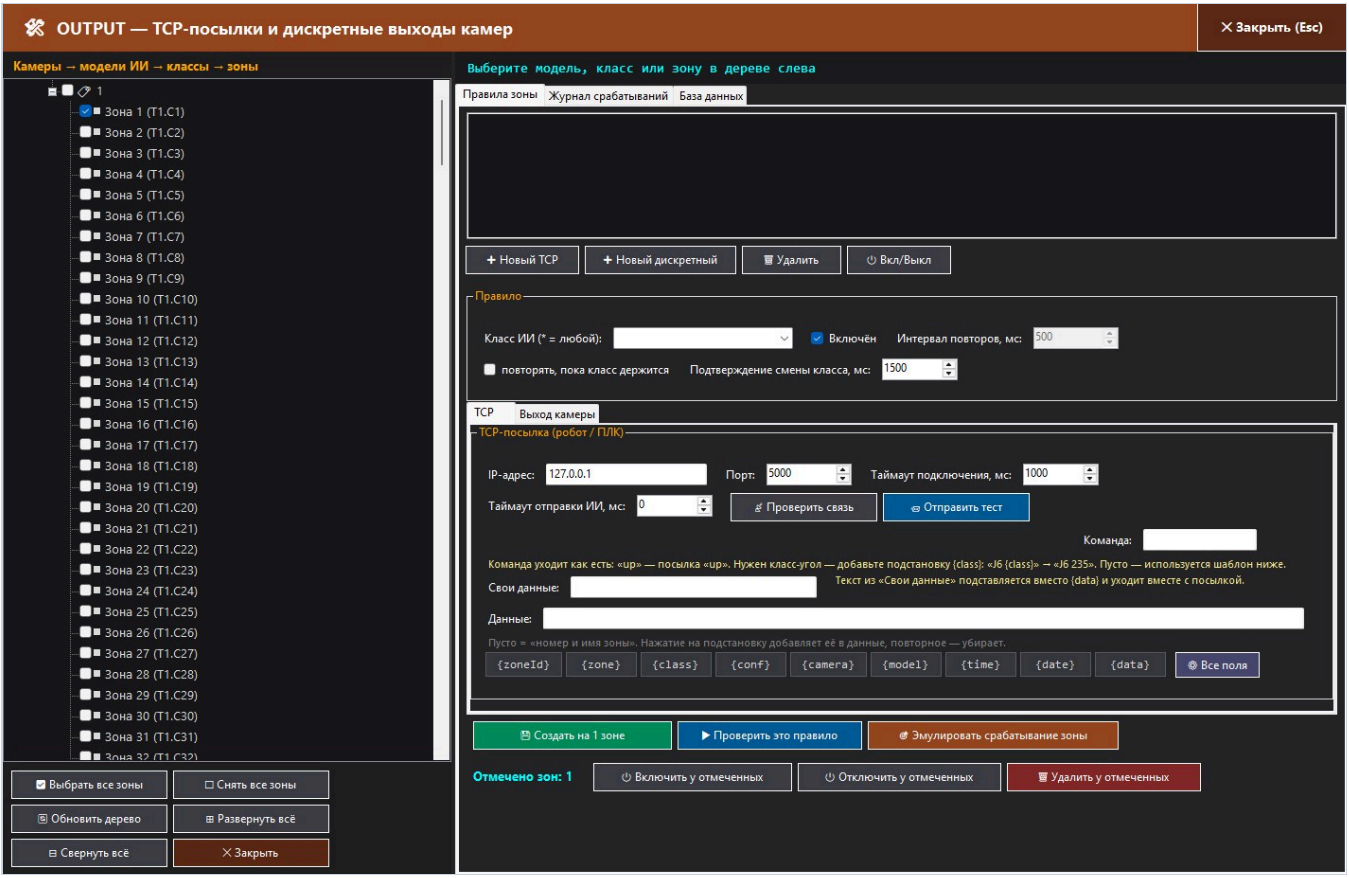


SICK picoCam — flash pin (photo-relay) on pins 5 (+) and 6 (−), 150 mA continuous.

- Output — "dry contact" of the photo-relay: it does not provide its own voltage, only closes the circuit. An external power supply is required.
- Testing with a multimeter in Ω mode: OUTPUT OFF — open circuit (OL), OUTPUT ON — hundreds of Ohms. The buzzer is silent in both cases, see the numbers. If nothing changes, reverse the probes; the photo-relay is polarized.
- Load: +24V → lamp → pin "+", pin "-" → GND. For HIKROBOT, the limit is 25mA – any signal exceeding 27mA will cause problems; reduce the power supply or use an intermediate relay. For SICK, the power consumption is high, typically 20-30mA per lamp, which can be connected directly.

7.2. OUTPUT Window: Zone Rule

The "**OUTPUT**" button on the right panel (or right-clicking on the area → "Create a rejector for this area") opens the "OUTPUT – TCP packets and camera discrete outputs" window. On the left, there is a tree structure "`Cameras → AI models → classes → zones`" with checkboxes on the zones; on the right, there are tabs for "**Zone rules**", "**Trigger log**", and "**Database**".



OUTPUT Window: list of zones on the left, zone rule list, "Rule" group, and TCP tab with settings.

1. Select a zone in the tree (the title on the right will show the path: `HD USB Camera → model_5 → 1 → Zone 1 (T1.C1)`). Click **+ New TCP**, **+ New Discrete**, or **+ New Audio**.
2. In the `Rule` group, select the `AI Class` that should be triggered (for ZONE-PAIR, this is either `anomaly` or `normal`; `*` – any class), check the `Enabled` box, and, if necessary, use the `Repeat until class holds` option with a repetition interval, and the `Class Change Confirmation, ms` (default: 1500 – the new class must remain in effect for that long for the change to be recognized).
3. Please fill out the **TCP** (Section 7.3), **Camera Output** (Section 7.4), or **Audio** (Section 7.5) section.
4. Verify the chain without waiting for a defect: `Verify this rule` (sends a package or triggers an output), `Simulate zone activation` (runs all zone rules through a combat queue with anti-jitter and logging).
5. Click "**Create for 1 zone**". If multiple zones are selected, the button changes to "**Create for N zones**"; on the model node, it says "Save to entire model", and on the class node, it says "Save to class". At the bottom: "**Enable / Disable / Delete for selected**".

One zone can contain multiple TCP rules (different receivers), no more than one discrete output rule per class, and no more than one audio stream per class. The rule is stored in the database and operates from the inference stream: rules are stored in memory, a search for the zone's address is performed per frame, and the camera's sockets and SDK are handled by a separate dispatcher thread to prevent the analysis from waiting for the network.

7.3. TCP Packet to Robot or PLC

The program functions as a TCP client: it connects to the specified **IP address** and **port** of the receiver, maintains the connection, and reuses it. If the connection is lost, it attempts to reconnect and resend the data once. The data is transmitted in UTF-8 encoding and includes a carriage return and line feed character at the end.

Field	Purpose
Connection timeout, ms	Default: 1000, 100...10000.

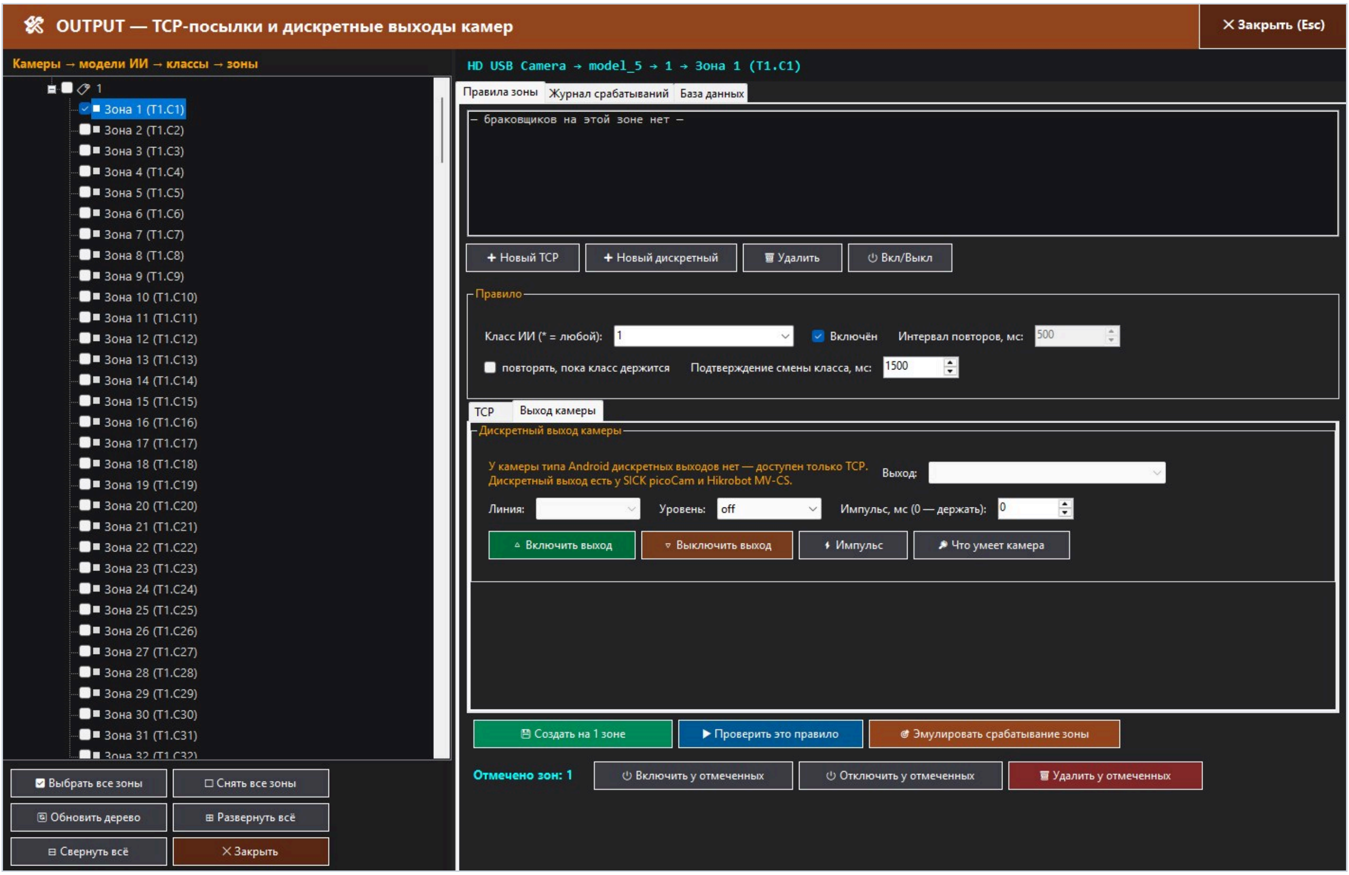
AI sending timeout, ms	Minimum pause between automatic rule transmissions; 0 = no pause. Even changing the class requires this pause.
Command	The word is transcribed exactly: "up" – the "up" package. A class angle is needed – add a placeholder: "J6 {class}" → "J6 235". Empty – uses the "Data" template.
Own data	Placeholder text that can be inserted in place of `{data}`.
Data	Template for a package. Empty = "zone number and name" ({zoneId} {zone}). Clicking on the placeholder adds it; clicking again removes it; "All fields" creates a structured template.

Substitution	Meaning	Substitution	Meaning
`{zoneId}`	Zone number (1, 2, 3...)	`{camera}`	Camera name
`{zone}`	Zone name (editable in the database)	`{model}`	AI model name
`{class}`	Class the model identified	`{time}`	Time HH:MM:SS.ms
`{conf}`	Confidence, % (using a dot as a decimal separator)	`{date}`	Date YYYY-MM-DD
`{data}`	Text from "Own Data"		

Example of a structured template: `zoneId={zoneId};zone={zone};class={class};conf={conf};camera={camera};model={model};time={time};date={date};data={data}`. Numbers are always in a consistent format to prevent parsing errors in PLCs.

- `Check connection` only opens the socket and reports either "connection established" or "no response after N ms".
- Send Test actually sends a string; the result is displayed next to the button, not within the window.
- **Confirmation.** Without the "repeat" checkbox, the rule sends one message upon class change and repeats it until the recipient responds with the line `ok <message>`. With the checkbox, the message is repeated at regular intervals while the class remains active.
- "Last command wins": If the model has generated multiple results while the operator is busy, the last one will be executed. The queue is limited to 512 tasks.

7.4. Discrete Camera Output



"Camera Output" Tab: Output method, line, level, pulse, manual check buttons. Discrete outputs are not available for USB cameras – only TCP is accessible.

Field	Purpose
Output	Method: Auto (SICK – Flash pin (constant level) or GPIO; HIKROBOT – UserOutputValue or LineInverter (strobing off)). The MV-CS020/016-10GC UserOutput node does not have a UserOutput – the program will automatically select LineInverter.
Line	Flash pin / GPIO1...6 (SICK); Line1 (output), Line2 (bidirectional) (HIKROBOT); Line0 – only input.
Level	true / false / off. "off" – rule is active, but the output is not affected.
Pulse, ms (0 – hold)	0 – level is held until the class changes; otherwise, a pulse of 10...5000 ms with return.

The buttons **Enable Output**, **Disable Output**, and **Pulse** are checked using a tester or lamp; **What the camera can do** queries all methods and prints a report. Manual tests are written in the same log.

> **Multiple zones on one output.** If multiple zones are connected to a single pin, the output is treated as a logical "AND": HIGH only when all zones request HIGH, and LOW if any zone requests LOW. This is represented in the log as "zones on output N require LOW k → HIGH/LOW". Thus, a single output can be created from hundreds of zones on a circuit board.

> **Important.** The SICK flash pin is the same for both the discrete output and the backlight: if you connect a pulsed lamp to it, use GPIO. The anti-bounce settings are 500 ms by default (minimum 200), and no event occurrences are logged.

7.5. Audio Alert: Custom sound for each zone

The third type of rule involves audio alerts from the computer running the software. The operator doesn't need to look at the screen; instead, they hear an alert when a zone enters a defective state, and different sounds in different zones indicate the specific location. This audio alert is available for any cameras, including USB and RTSP cameras, which do not have discrete outputs.

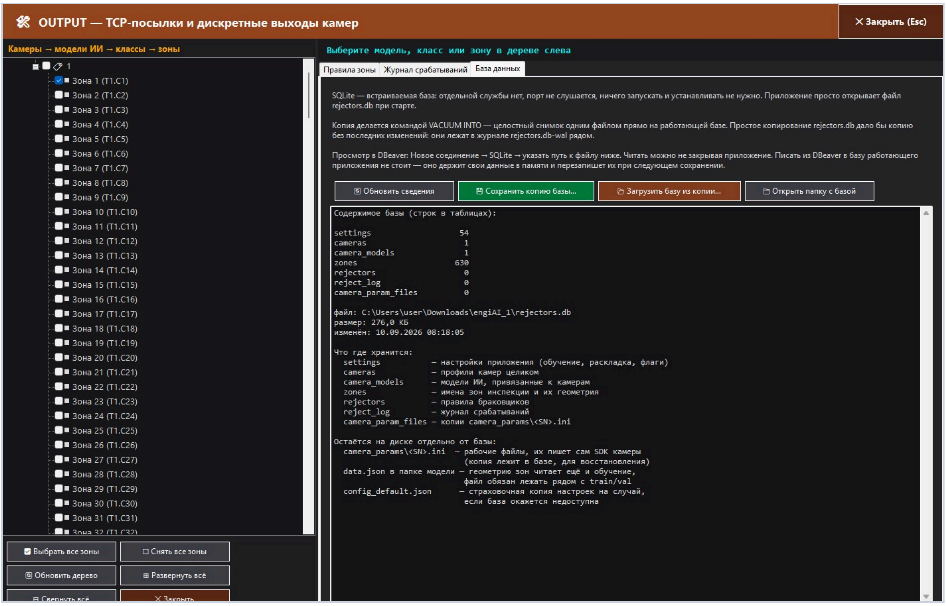
1. Select a zone in the tree and click "**+ New Sound**". In the "Rule" group, specify the class (e.g., "`anomaly`") and check the "**Enabled**" box.
2. On the "**Sound**" tab, select a signal from the list. There are ten built-in signals: short, double, triple signal, low and high tone, ascending and descending, alarm siren, long horn, bell. The "**Play**" button plays the selected signal.
3. Your own sound: click "**Upload your file**" and select WAV, MP3, or WMA. The file is copied to the "`sounds\custom`" folder next to the program (the USB drive can be removed), and it appears in the list with the label "your own". The "**Sounds folder**" button opens this folder: you can directly place files there.
4. Set the **pause between sounds, ms** to a value between 1000 and 10000. During this time, the sound will not repeat after the signal, even if the zone class changes again: with a 10000 ms pause, the operator will hear no more than one signal every 10 seconds.
5. Click "**Create for 1 zone**" or select the zones with checkboxes and create the rule for all of them at once. Assign a different sound to each zone: repeat the process with a different signal for each zone.
6. The checkbox "**Yellow (muted by clicking) zones are silent**" is enabled by default: zones that are muted by clicking on the "Inspect" tab (section 5.5) do not produce sound or trigger TCP or camera output throughout the model.

Behavior	How it works
Without the "repeat" checkbox.	The sound occurs once after a confirmed change in zone class (after "Zone Class Change Confirmation"). As long as the class remains unchanged, there is silence.
With the "repeat until the class is still" checkbox.	The signal repeats according to the "Repetition Interval," but not more frequently than the pause between sounds, as long as the area remains within the specified class.
Several areas in the defect simultaneously	Each zone produces its own sound; signals from different zones do not interfere with each other, but only the signal from a single zone is silenced.
Journal	Each trigger is recorded in the "Trigger Log" as a line stating "Sound: ..."; if the file is not found or the audio device fails, the line indicates an error, and the program continues to function.
Verification	**Verify this rule** immediately plays the sound; **Simulate zone activation** plays the sound along with TCP and outputs through the combat queue.

> **Advice.** Use a siren or a triple signal for the most important areas (connectors, power elements), and short signals for the rest. This way, a single sound indicates whether to immediately move towards the line. The default sound setting for the entire model is created by selecting the model or class node in the tree: one signal for any defect.

7.6. Trigger Log and Database

The "**Operation Log**" tab displays the most recent 300 entries in rows of the format "`HH:MM:SS.mmm ✓/✗ Camera · Model · Class · Zone → Action [message]`", and it is updated in real-time. The "**Refresh**" and "**Clear Log**" buttons.



"Database" tab: where the data is stored, creating a copy, and restoring the database with a single file.

Tab **Database**The program stores all data in a SQLite file `rejectors.db` Next to the executable file, there is no separate service running, and the port is not active. No installation is required. Tables: "settings" (settings), "cameras" (camera profiles) camera_models (models on cameras), zones (zone names and geometry), rejectors (rules) reject_log (journal) camera_param_files (copies of camera settings), camera_presets (camera settings configurations). **Save a copy of the database...** It creates a comprehensive snapshot of the VACUUM INTO process directly on the active database. **Load the database from a copy...** It restores the database, but first saves the current version. You can view the database in DBeaver without closing the program; however, you should avoid modifying it from outside the program.

8. Other Modes and Tasks

Standard models: normal / anomaly and any classes

The same ROI editor also works for standard classification: pills, caps, labels, and parts in the tray. The classes are defined in the "New AI Model" dialogue (default: normal and anomaly, with Latin characters), and the zones are assigned to the items. The "**Take a picture**" button with numbers 1...50, the "**Create images**" series, either from the camera (frame/shot) or from the encoder, and the "**Synthetic data for AI**" panel: precise rotations of 90°/180°, a full 360° rotation with a step and range, a mirror, "classes by degrees" (each angle in its own class – the rejector sends the angle via TCP), and brightness and contrast ranges. Training is performed with full Ultralytics augmentations. The confidence threshold for these models is set in the same place as for ZONE-PAIR – on the Inspect tab.

Multi-model inspection (slots 1–20)

With a single camera, you can manage up to 20 models: **Add ONNX to the slot**, then **ROI → Add a class to the selected model** – each model in the slot has its own zone labeling, and the results from all models are displayed on a single frame.

> **ZONE-PAIR slots are typically not required.** One model already covers the entire board: each zone has its own pair of classes, so the need to "merge similar classes" that previously justified having multiple models, no longer exists. A typical setup involves one ZONE-PAIR model per board, with the slots empty.

When multi-model inspection is actually used:

Situation	How to configure
Different tasks on a single frame	ZONE-PAIR checks the layout and the presence of elements, while the standard model in the second slot marks the item with a label, sticker, color, or key direction. The zones of the second model are added separately using "ROI → Add class to selected model".
Product without a fixed grid	Tablets, covers, and components in the tray: multiple product types in standard models. Similar types are arranged in different slots to prevent confusion.
Two different boards in one frame	Two ZONE-PAIR models in different slots, each with its own reference and grid. The live conveyor works with the reference of the main model.
Transition period	The old, trained model remains in the slot while the new ZONE-PAIR gathers statistics; both are visible on the same frame, and the rejector rules are transferred as trust increases.

The threshold, analysis speed, yellow zones, and rejector rules apply to all camera models. The "**REMOVE MODEL**" button clears and resets the slots, so it asks for confirmation; "**Remove selected**" removes one model from the slot, but its files and zones remain in the database.

Common models

The "**GENERAL MODELS**" button saves a set of "camera → model" entries under the name: "**Create from current models**", "**Apply to all cameras**", "**Update from current**", "**Rename**", "**Delete set**". Switching the stand from eight cameras to another product is a single selection. Deleting the set only removes the grouping: folders, ONNX, zones, and rules remain. The zones belong to the pair "camera + model": one model on eight cameras results in eight independent annotations.

Encoder and rejector on the conveyor

For lines with an encoder on the COM port (115200, ESP32): connect **CONNECT ENCODER**, auto-connect, position, image capture step, and correction, **Set zero + CAPTURE**. Capture for training and analysis can be done "via the encoder" - the frames are evenly spaced along the product's path. The old rejector, with a time delay and anti-spam function, sends the

REJECT command to the encoder controller.

Video and image recording

Settings tab: **START VIDEO RECORDING** from the active camera to a file named `record_date_time.avi` (MJPG, actual camera FPS). Training images: `real_….jpg`, inspection images: `inspect_….jpg` with marked areas.

9. Settings, console, requirements, and license

Settings

- SAVE TO FILE / LOAD FROM FILE – all settings (camera, models, rules) in JSON format, which is convenient before reinstalling or transferring to a second computer. The source of truth is the database, and the JSON file serves as a backup in case the database becomes unavailable. All changes to the settings are automatically saved.
- RESET TO DEFAULT – This action is irreversible: it removes the configuration and window layout.
- Interface language: 14 languages; the main window switches immediately, and the other windows switch upon the next opening.
- Position, size, and monitor of all windows, and the position of dividers – are stored in the database; when the monitor is turned off, the window returns to the visible area.

Console

The "Camera" and "Trigger" checkboxes are disabled by default, and the corresponding messages are displayed (in trigger mode, camera messages go to the "Trigger" section); all data is written to the `logs\engiai-*.log` files before any filtering is applied. The console is designed to operate for weeks without intervention: up to 800 lines are displayed on the screen, with 1000 old lines being removed; new lines are collected in batches of 4 every second, rather than one at a time; downward scrolling is enabled until you manually scroll upwards; the console has no sound. Changes to zone classes are written no more than 10 lines per second (the rest is stored in a counter), the log file is 20 MB in size, and files older than 14 days are deleted upon startup; the log of events in the database stores the last 20,000 entries. Unprocessed errors are located in `crash.log`, and the reasons for forced termination are located in `runtime.log` next to the exe file. If something is not working correctly, check this section first.

Requirements

Component	Requirement
Operating System	Windows 10/11 x64.
Inspection Video Card	NVIDIA with CUDA driver; CUDA 13 and cuDNN 9 libraries are included with the software. Inspection runs on the processor automatically without a video card.
Training	Python 3 (with the "Add python.exe to PATH" option during installation) and the ultralytics, onnx, onnxslim, onnxruntime packages – the program automatically installs them during the initial training process, requiring an internet connection. The GPU used for training is determined by PyTorch; without CUDA, training will proceed on the CPU, but at a slower pace, with the same result.
SICK Cameras	IDS uEye driver; the camera must be visible in IDS Camera Manager.
HIKROBOT Cameras	MVS SDK (library included with the software); for GigE, a network card with jumbo frame support is required.

License and Trial Mode

Upon startup, the program connects to the engi.live license server. Only hashes of the hardware identifiers and version numbers are sent, without serial numbers or user names. If the server is unavailable, but a license has been cached, the program operates using the cached license (for offline environments). Without a valid license, a 20-minute trial mode is activated: a "TEST MM:SS" countdown appears in the window title bar and in the AI status bar. The "BUY" button on the Settings tab opens the purchase page. When a valid license is active, the countdown is not displayed.

10. Quick Checklist and Fault Finding

Zero-based checklist for the board

1. CAMERAS → "Add Cameras" → Select your cameras → Add.
2. Camera Tab: Resolution → "Auto-configure for AI (per frame)" (or "Set parameters on USB") → "Save parameters" → "Save as..." settings. Optionally, "Train in grayscale."
3. AI Model → "Create new AI model" → Name.
4. "ROI → Add to class (training model)" → Draw a frame around each component → Zone size (e.g., 256) → Enter. For hundreds of identical components, draw a frame around the entire board and use a grid.
5. Checkbox "Mark zones to product by markers" → "Mark reference" → Draw around the component area on the frame (not the inspection zone) → "✓ Search zones by selected area."
6. "ZONE-PAIR: Capture by zones" → "Capture." Repeat on 3-5 good boards. Disable unnecessary zones in advance by clicking on the frame (red).
7. «ZONE-PAIR: zone anomalies» → «Create»: The element's shift and rotation, along with the base, are already included. If necessary, solder pads in the "reference with zones" section.
8. TRAIN AI MODEL (Image Size 128, s, 50 epochs) → wait for 100% completion → best.onnx is automatically assigned.
9. Inspect → «ZONE-PAIR: zones ↔ model» → verify → threshold. Choose either "by camera" or a trigger. For a moving board, use "Live Conveyor".
10. OUTPUT → select zones → rule for the anomaly class (sound, TCP, or camera output) → "Simulate zone activation" → "Create on N zones".

If something goes wrong:

Symptom	What to check
Camera does not appear in the list	Power and cable; for SICK — IDS Camera Manager and IP from the adapter's subnet; for HIKROBOT, wait up to 25 seconds after an emergency shutdown and "Reconnect"
Image is dark, gray, or has incorrect colors	Camera → "Auto-configuration for AI"; for color SICK — Bayer pattern and white balance; for USB — "Standard", then "Lock parameters"
Frames are dropping, network is red	GigE: packet size of 1500 or jumbo on the network card, inter-packet delay; lower FPS or fewer frames; separate network card for the camera
"Component not found" / image missed	reference not detected or captured under different lighting; few reference points – enlarge the alignment region with details; the area contains a table or conveyor; increase the search period
Zones are shifting to another board	the board is searched by the inspection zones: re-establish the reference with the alignment region around the board (section 4.4); automatic alignment model
Defective boards from a different batch are missing	marking on the enclosures is different: add normal images from this batch; do not create synthetic scratches thinner than 3 px (section 4.6); check that the defect is a shift, not a color
Areas marked with NO_ZONE_CLASS	the grid changed after training; return the grid or create a new model and reshoot

All zones are red on a good board	another payment or different position without a live conveyor; the threshold is too high; the photos were taken under different lighting; mixed color and grayscale files
The model does not detect any movement of the element.	recalculate px/mm (section 1.3): with a small scale, the shift should be less than 5 px; reduce the field of view or use a camera with higher resolution; do not increase the shift
Learning is declining or progressing very slowly.	Device = CPU if no NVIDIA graphics card is present; reduce Batch Size; enable Console output; verify that Python is in the PATH.
The inspection system is not functioning correctly.	Is best.onnx selected? Is the speed source enabled (via camera / timer / trigger)? Is the "DO NOT USE AI" option unchecked? Is the "AI Enabled" checkbox selected on the camera?
The "rejector" rule is silent.	"Create on N zones" checkbox: checked; rule class matches the model's answer (anomaly / normal); for exit, the level is not "off"; USB cameras have no outputs, only TCP.
The sound isn't working.	The rule has been created and enabled; the file is located in the "sounds" (or "sounds\custom") folder; is there a pause between the sounds; when "repeat" is disabled, the sound only plays when the class changes; Windows volume and output device.
The TCP packet is sent only once.	The "repeat" function requires a response of "ok <package>" from the receiver; either check the "repeat while class is active" box or configure the response on the PLC side.

engiAI Ethernet System. The names of buttons and fields are as they appear in the program on your language (screenshots are from the Russian interface). The screenshots were taken on the "model_4" model (two 256x256 zones: capacitor C30 and bridge BR1), a 3264x2448 USB camera, and the model was trained in grayscale. Version 3, September 2026: example with two zones, creating model folders, alignment region, synthetic data "element with base" with "before / after" images, color palette and line thickness, red zones, console.