Sentinel — Getting Started

From first start to your first trained model

Larix Technologies Kft. · 2026

Sentinel — Getting Started

This walks you through everything from launching the app for the first time to training and testing your first AI model. The bundled **Sample - VisA PCB1** image set is used in the screenshots so you can follow along on a fresh install without supplying your own pictures.

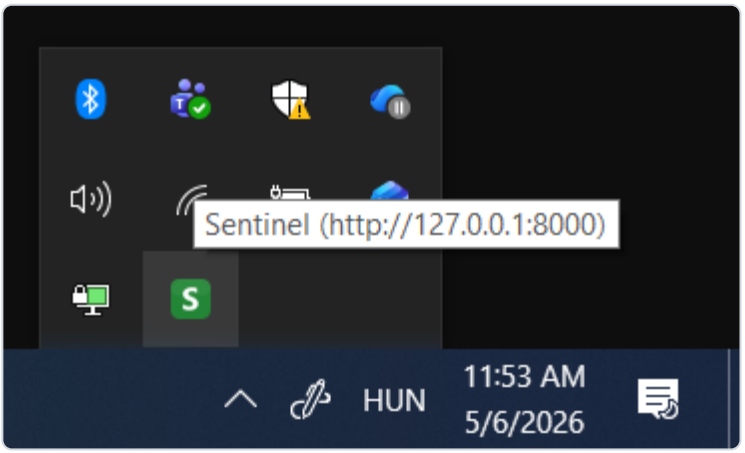
Installation is covered separately — this guide assumes Sentinel is already installed.

1. Starting Sentinel

After installation, Sentinel runs as a tray application — there is no main window.

- Double-click **Sentinel** in the Start Menu, *or*
- Double-click `Sentinel.exe` in `C:\Users\<you>\AppData\Local\Programs\Sentinel`, *or*
- If you ticked "*Start Sentinel automatically when I log in*" during install, it will already be running

A green Sentinel icon appears in the Windows system tray (bottom-right, near the clock — you may need to expand the hidden-icons arrow `^`).

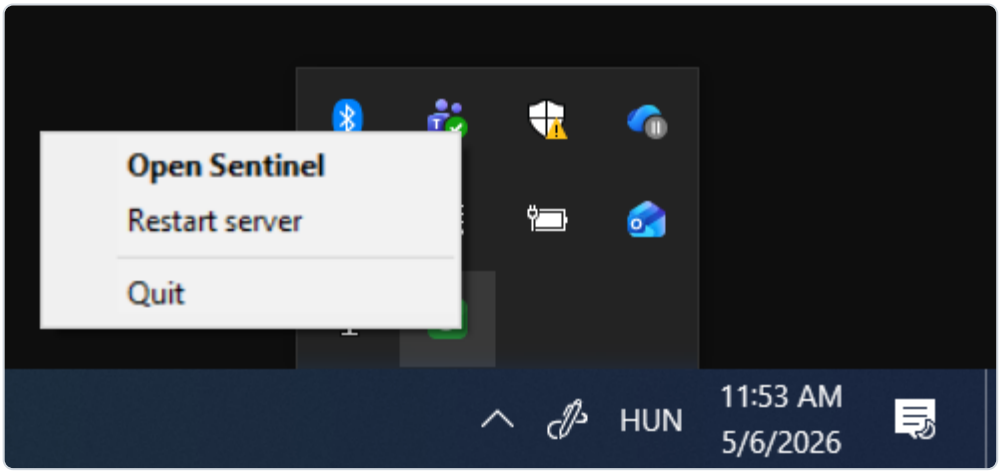


The first start takes 10–30 seconds while the embedded server initialises. Subsequent starts are faster.

© 2026 Larix Technologies Kft.

2. Opening the Web UI

Right-click the tray icon and choose **Open Sentinel**. Your default browser opens to the application at `http://127.0.0.1:8000/`.



If the page does not load on the first try, wait a few seconds and choose **Restart server** from the same menu, then **Open Sentinel** again.

Reach the UI from another device on the same network: open a browser on the other machine and navigate to `http://<sentinel-host-ip>:8000/`. Find the host IP by running `ipconfig` on the Sentinel machine and looking for the IPv4 address of your active adapter (e.g. `192.168.1.28`). Windows Firewall will prompt you the first time another device connects — accept on the *Private network* profile.

---

3. Activating your license

Sentinel is licensed per machine. On the very first launch you will see a banner at the top of the app that says:

> **System is not activated. Send your machine ID to your vendor to receive a license.**

Below the banner, the **Machine ID** is displayed (a short alphanumeric string unique to this PC's hardware). Until a valid `license.lic` is uploaded, training, inspection and the test bench are disabled — only browsing and login work.

How to request a license

Send an email to `info@larix-technologies.com` with the following:

- **Subject**: `Sentinel license request — <Your Company Name>`
- **Body**:
  - Your **company name** (will appear in the activated-for label inside the app)
  - The **Machine ID** copied from the banner
  - Optional: number of users / requested validity period / contact person

© 2026 Larix Technologies Kft.

You will receive a `license.lic` file by reply, typically within one business day.

How to activate

1. Save the attached `license.lic` somewhere convenient.
2. In Sentinel, click **Upload license** on the activation banner (or under **Admin** → **Settings** → **License**).
3. Pick the file. The banner disappears and "Activated for `<Your Company>`" appears in its place.

The license is bound to this machine's hardware. If you move Sentinel to a different PC, request a new license — the old file will not work on the new machine.

Renewing

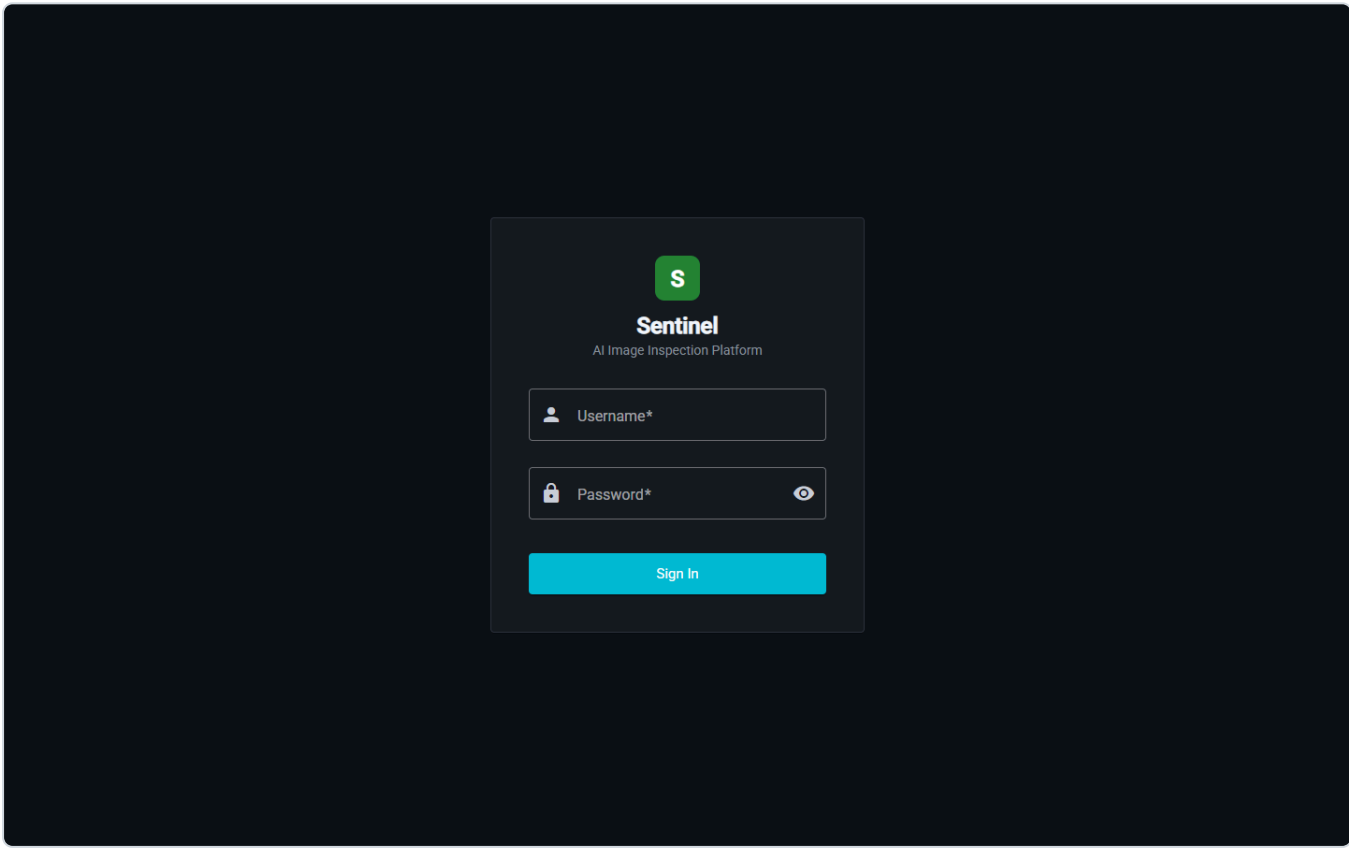
Two weeks before expiry, Sentinel shows a yellow banner:

> **License expires in N day(s) — contact your vendor to renew.**

Email the same address with the subject `Sentinel license renewal – <Your Company Name>`. Upload the renewal file via **Upload renewal** on the banner.

---

4. Logging in



Default credentials on a fresh install:

- **Username**: `admin`
- **Password**: `admin`

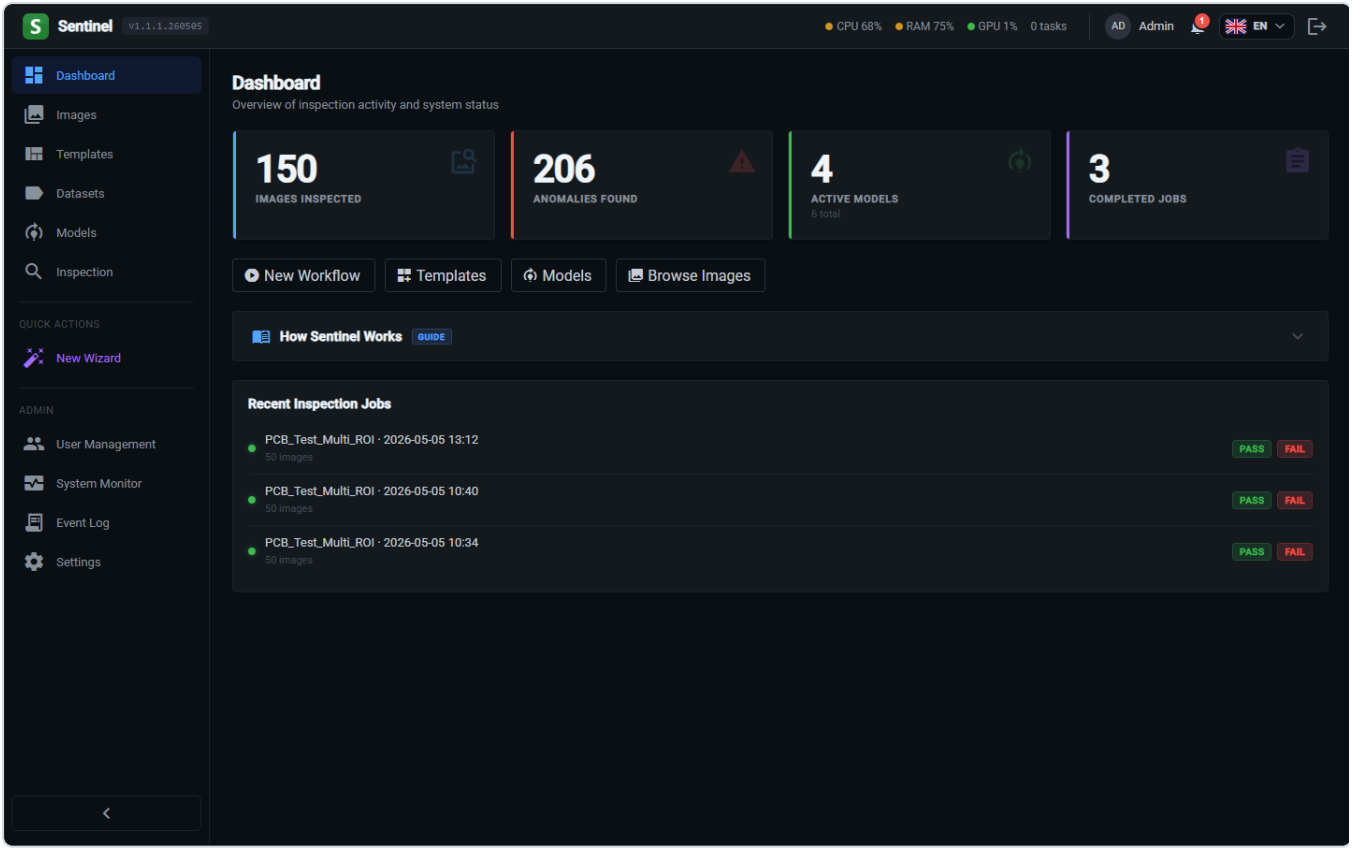
Change the password immediately under **Admin** → **Users** after first login.

---

5. The dashboard

After login you land on the Dashboard. The left sidebar is your main navigation:

- **Dashboard** — overview and quick start
- **Templates** — define what you're inspecting and where
- **Datasets** — labelled image sets used for training
- **Models** — train, version, and publish AI models
- **Inspection** — run trained models on new image batches



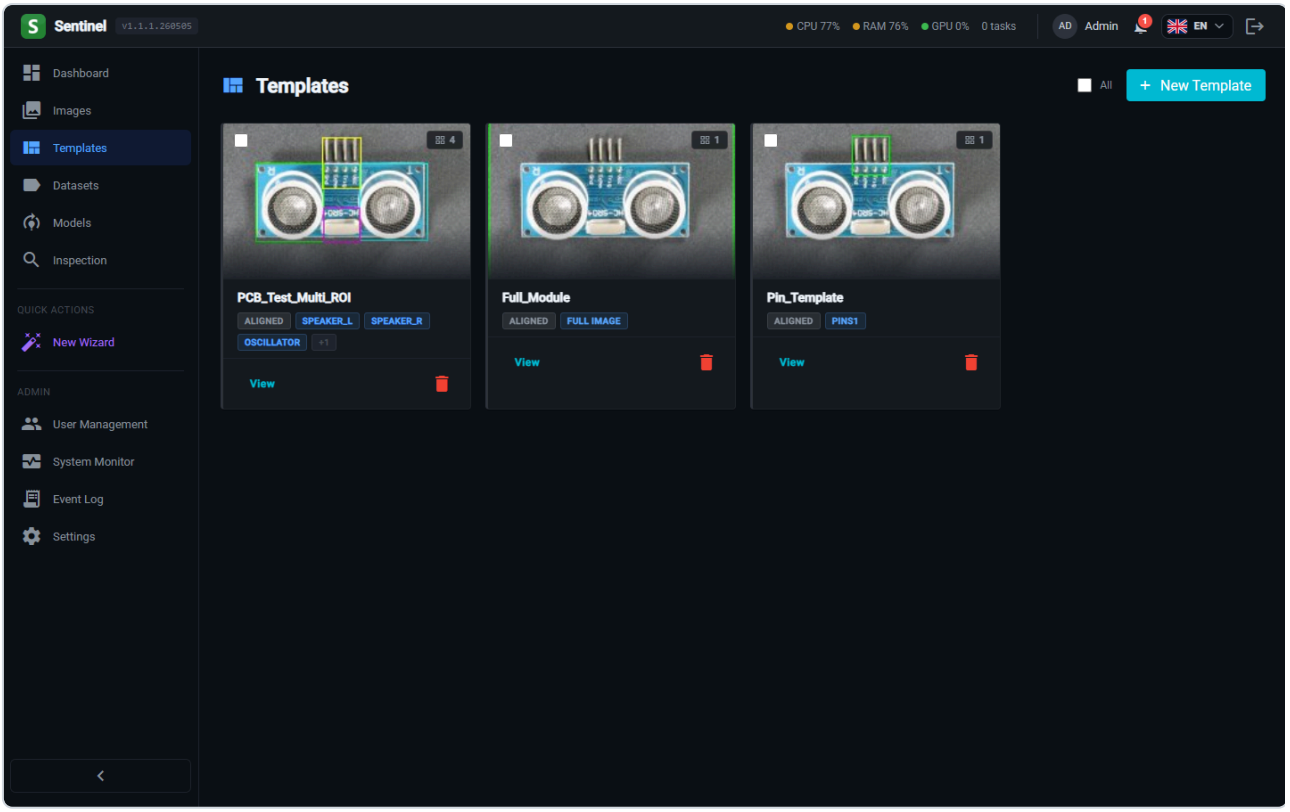
---

6. Creating a template

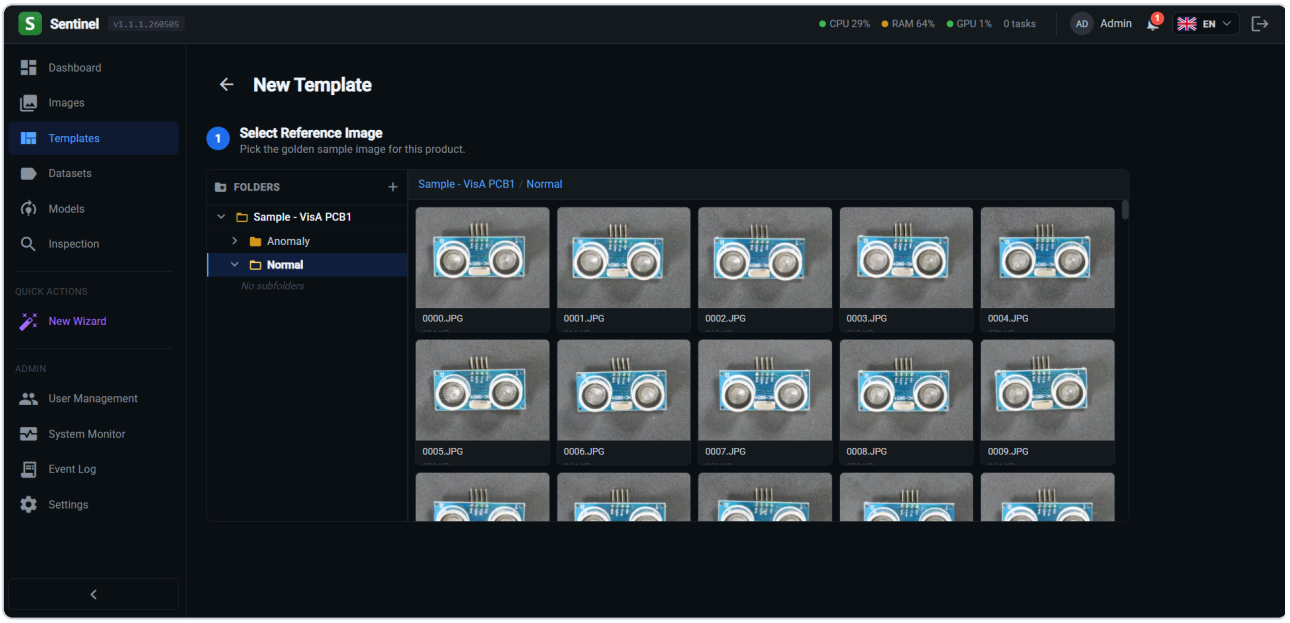
A **template** describes what a "good" product looks like and where on the image to look. Every dataset and model is anchored to a template region.

1. In the sidebar, click **Templates** → button **New Template** (top-right).

© 2026 Larix Technologies Kft.

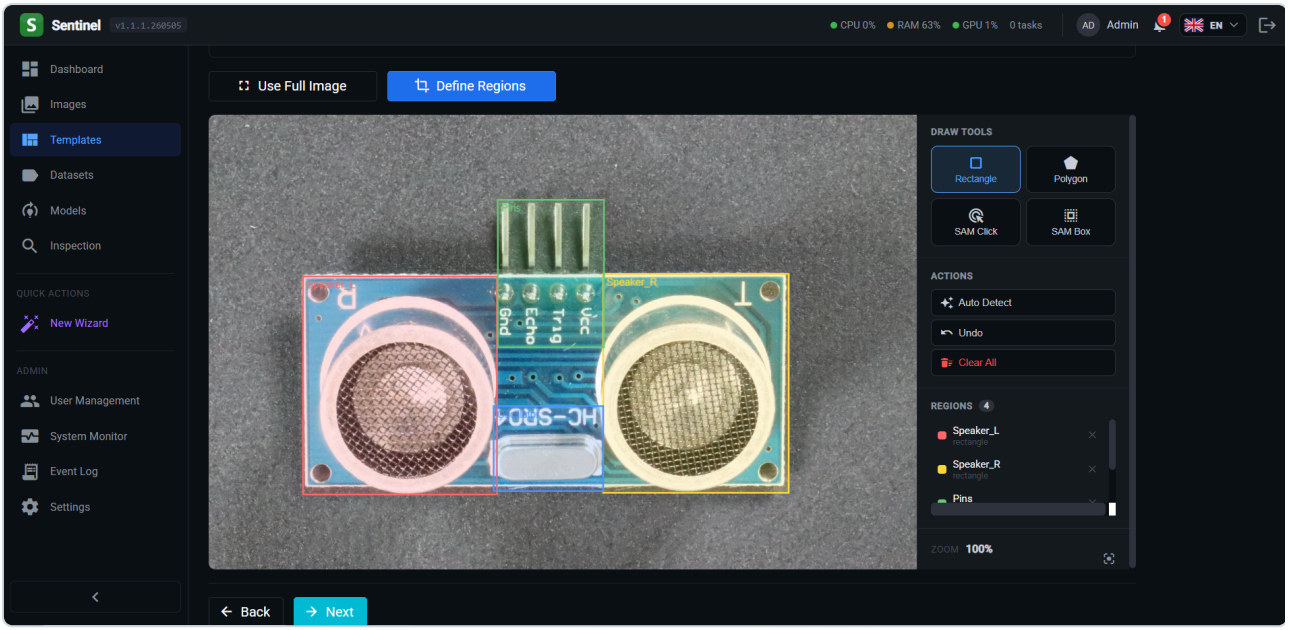


2. **Pick a reference image.** Browse to the *Sample - VisA PCB1* root → *Normal* folder → select any image (e.g. `0000.JPG`). Click **Next**.



3. **Draw regions of interest.** A region is the part of the image the AI will inspect. Click and drag on the image to draw a rectangle; give it a name (e.g. `Solder area`). Repeat for as many regions as you need. For testing, **one rectangle covering most of the board is enough**.

© 2026 Larix Technologies Kft.

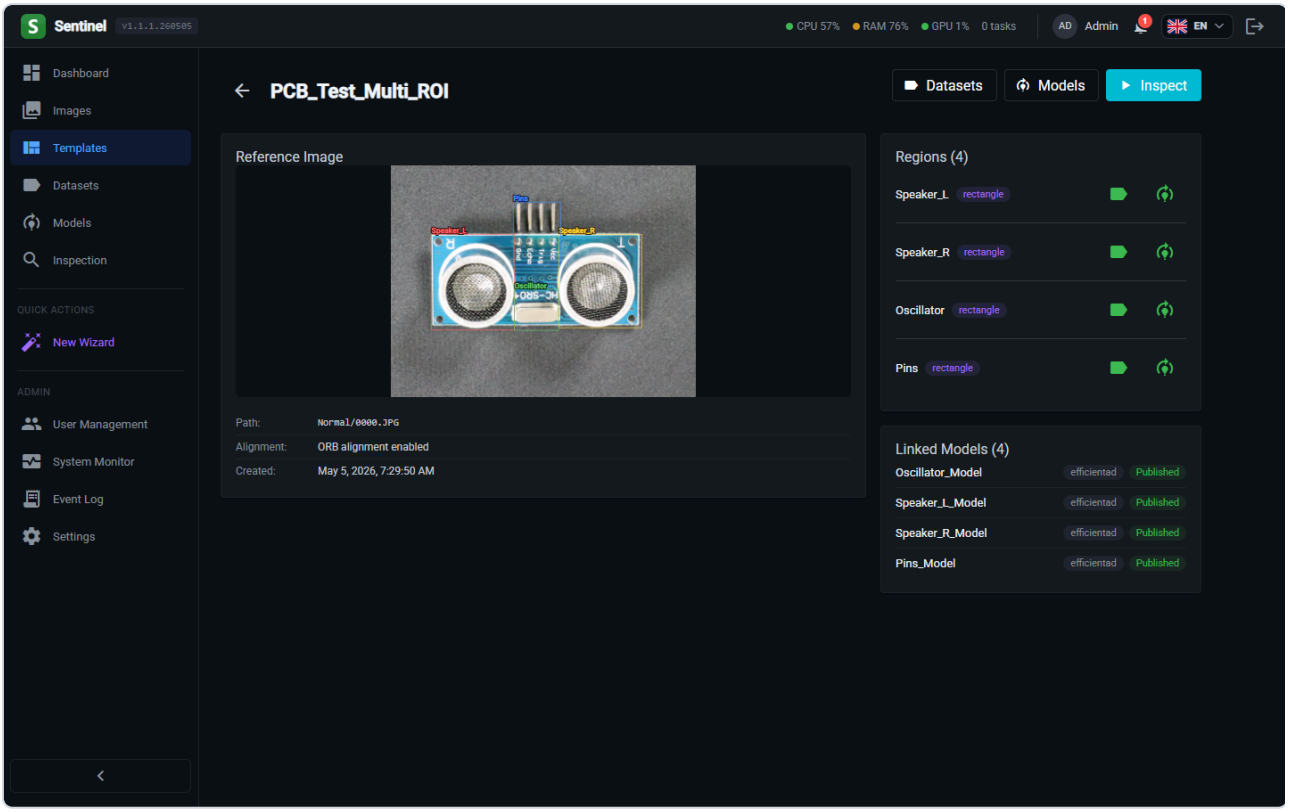


Region options at this step:

Option	What it does	When to use
Use Full Image	Treats the entire image as one big region — no need to draw anything.	Defects can appear anywhere; or you want one model that judges the whole product at once.
Rectangle ROI	Axis-aligned box. The fastest to draw and the most common choice.	Most components: solder pads, connectors, screen areas, label patches.
Polygon ROI	Click-to-place vertices, double-click to close. Crops to an arbitrary shape, masking everything outside the polygon to black.	Irregular outlines: cables, curved bezels, anything where a rectangle would include a lot of irrelevant background.
Region groups *(advanced)*	Tag multiple regions with the same group. One model trained on the group then applies to all members.	Repeated identical features — e.g. 8 coils, 4 corner screws, an array of solder pads. Saves you training 8 separate models.

4. **Image alignment.** Toggle **Enable ORB alignment before cropping** (in the wizard's settings panel). When ON, Sentinel automatically aligns each new image to the reference image before cropping the regions — using ORB feature matching plus a homography fit. Recommended for any product that may shift, rotate, or translate slightly between shots. Turn OFF only when you have a perfectly fixed jig where parts always land in the same pixel position.
5. Click **Save template**. You return to the Templates list with your new template.
6. Click your template — the detail page now shows the reference image with each ROI outlined in a distinct colour, plus a list of the regions on the right.

© 2026 Larix Technologies Kft.

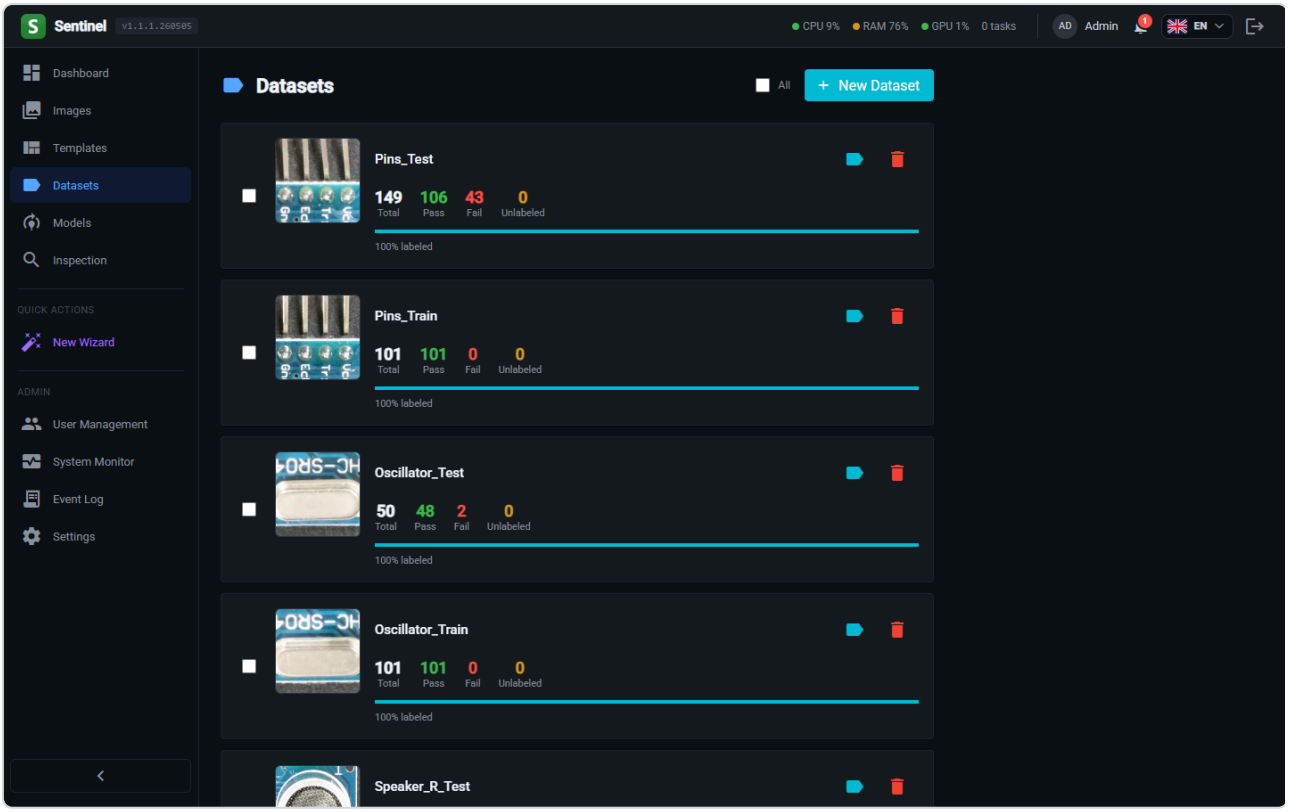


7. Building a dataset

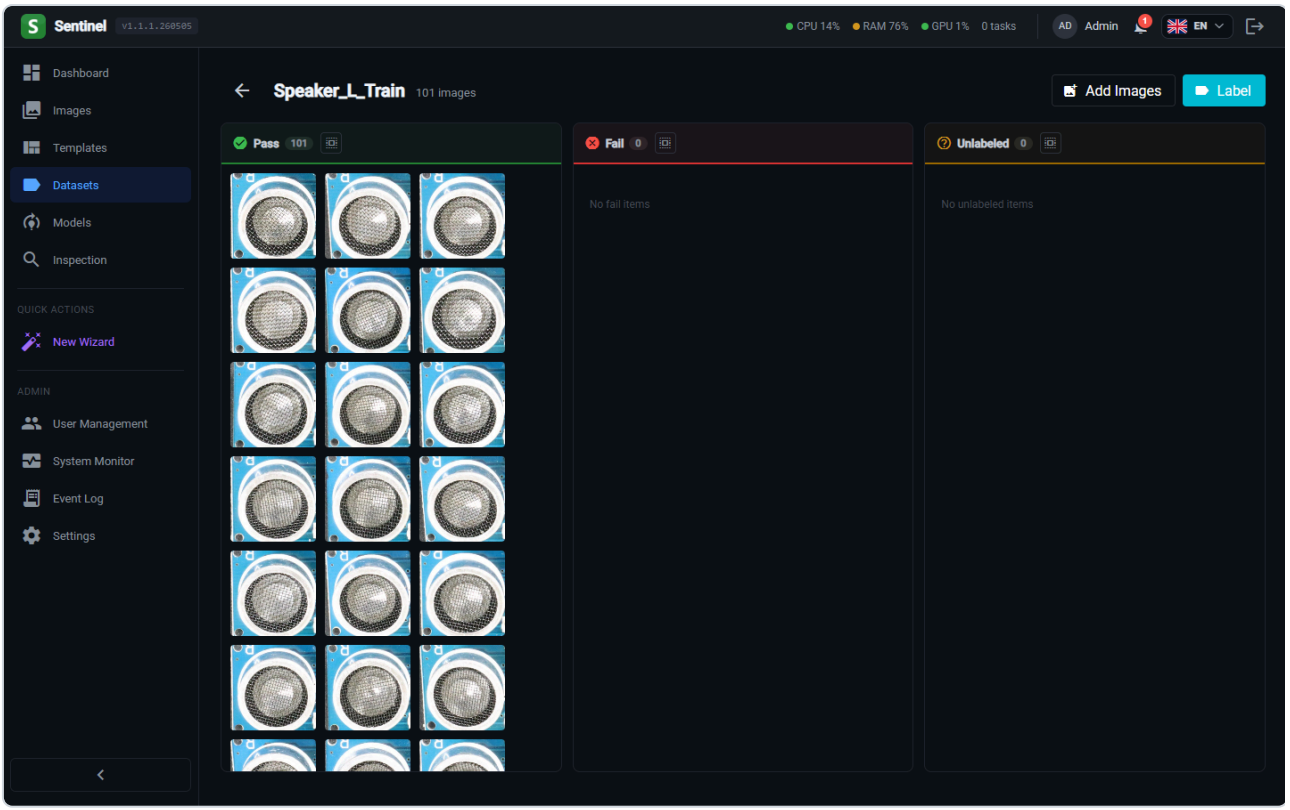
A **dataset** is a collection of cropped region images, labelled as Pass or Fail. The model trains on these crops.

1. From the template detail page, click the **dataset icon** (label tag) next to one of your regions. This opens the Datasets page filtered to that region.
2. Click **New Dataset** → name it (e.g. `PCB1 - main area`) → **Create**.
3. **Add images.** Click **Add Images**, browse back into *Sample - VisA PCB1*, select 30–50 images from *Normal* and 10–20 from *Anomaly*, click **Add**.

© 2026 Larix Technologies Kft.



4. Sentinel automatically aligns each image to the template's reference and crops out your region. A progress bar shows the work; usually a few seconds for this many images.
5. **Label the crops.** Click **Label** to enter the labelling view. Each crop appears in a 3-column layout (Unlabelled | Pass | Fail). Click a crop to assign it: **P** for Pass, **F** for Fail. Tip: keyboard shortcuts make this fast.



© 2026 Larix Technologies Kft.

6. Once labelled, return to the dataset list — the row shows your label counts.

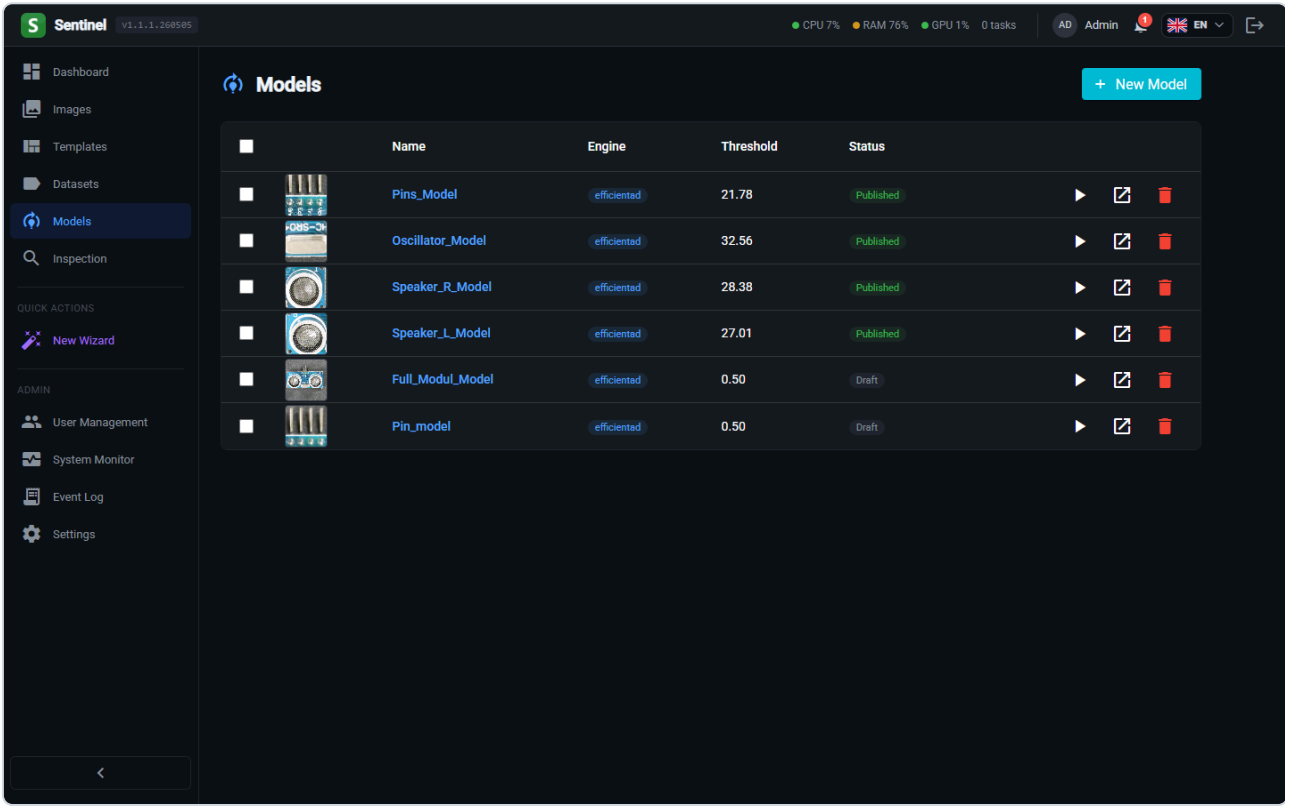
> **Note**: For *EfficientAD* (the model you'll train next) you only strictly need Pass labels. Fail labels are still useful for validation. ~50 Pass + 10 Fail is enough to demo.

> 💡 **Build TWO datasets per region — one for training, one for testing.**
>
> Repeat steps 1–6 to create a second dataset for the same region (suggested names: `<region> - Training` and `<region> - Test`). A natural split is roughly **70 % training / 30 % test**, with the test set drawn from images the model will *never* see during training.
>
> The test dataset is loaded into the **Test Bench** (Section 9) to evaluate how well the trained model generalises to unseen examples. Without a separate test set you can only judge the model on data it has already memorised, which always looks artificially good.
>
> For both datasets, mix Pass and Fail labels in roughly the same ratio so the test bench's precision/recall numbers are meaningful.

---

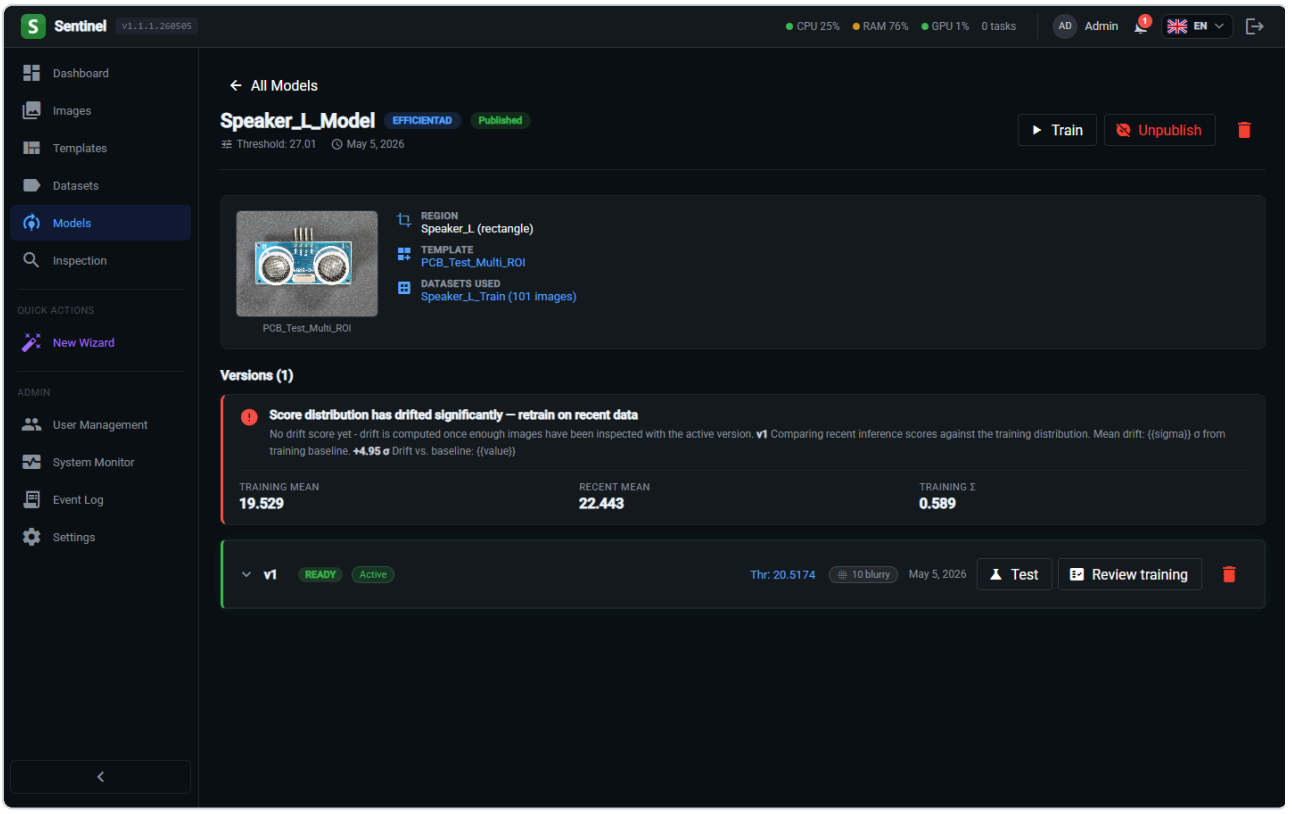
8. Creating and training a model

1. Sidebar → **Models** → button **New Model** (top-right).
2. Choose your **target region** (or region group). Pick the region you built the dataset for.



3. Pick the engine: **EfficientAD** (the only enabled card in this prototype build — others are greyed out).
4. Give the model a name (e.g. `PCB1 EfficientAD v1`) and click **Create model**.

5. You land on the model detail page. Click **Train new version**.



6. **Pick the dataset** you just built (the *Training* one — keep the *Test* one for the bench in Section 9). Leave training parameters at defaults for the first run — they're tuned to work out-of-the-box. Click **Start training**.

   What the training settings mean:

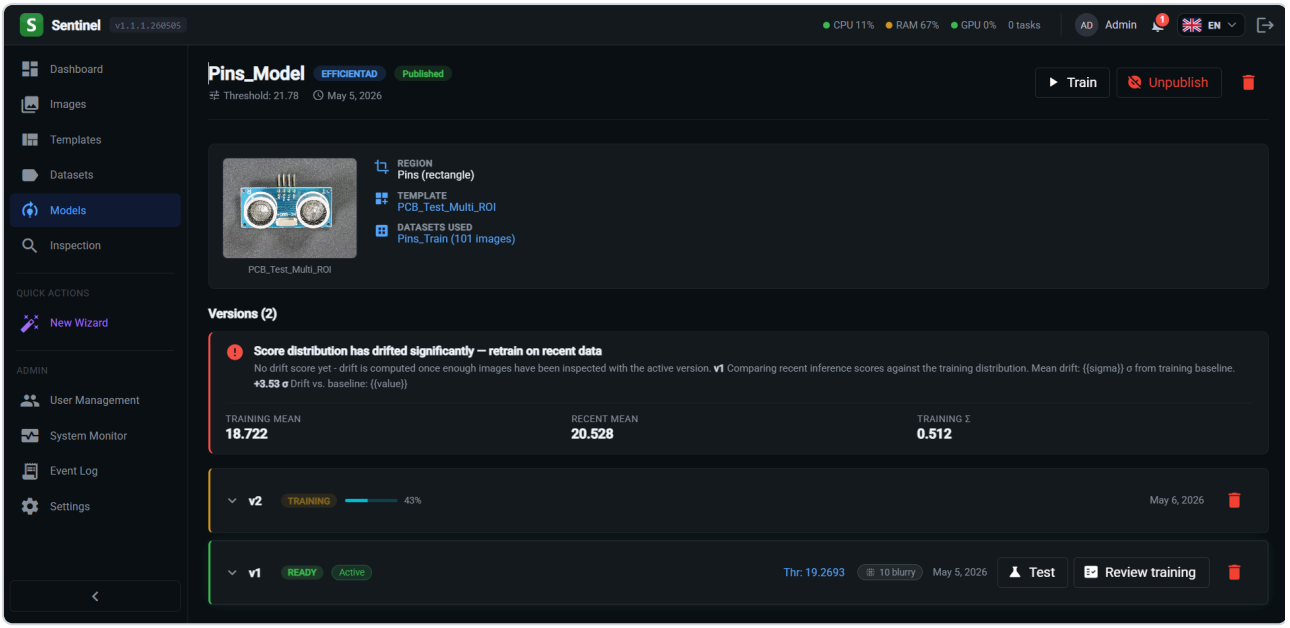
Setting	What it does	Default	When to change
Validation Split	Fraction of the training set held back to compute in-training metrics.	0.2 (20 %)	Raise to 0.3 if you have lots of data; lower to 0.1 with very small datasets.
Backbone	Pretrained CNN that extracts features from each image. **Wide ResNet50** = best accuracy / most VRAM, **ResNet50** = balanced, **ResNet18** = ~3× faster, slightly lower accuracy.	Wide ResNet50	Pick ResNet18 if training is slow or you're on CPU; pick Wide ResNet50 for the final production model.
Image Size (px)	Images are resized to this square before feature extraction. Higher = catches finer defects but slower.	256	Raise to 512 only for very tiny defects; 128 for fast iteration on a CPU.
Batch Size	How many images go through the GPU at once. Higher = faster on GPU but more VRAM.	8	Drop to 4 if you hit out-of-memory; raise to 16 if you have a high-VRAM GPU.
Subsample Ratio *(EfficientAD)*	Fraction of patches kept per image when building the memory bank. Higher = more representative bank, more memory + time.	0.1	Raise to 0.2–0.3 if your normal set is very varied.
Memory Bank Size *(EfficientAD)*	Total patches kept in the bank after subsampling.	10 000	20–50 k if your good products vary a lot (lighting, slight design changes).
N Neighbors *(EfficientAD)*	How many nearest patches to compare against when scoring. Lower = more sensitive.	9	Lower to 3–5 for sharper but noisier detection; raise to 15 to smooth scores.
Top-K Percent *(EfficientAD)*	Average the top K % worst patches when scoring an image. Lower = more sensitive to single bad patches.	1	0.5 if you want to flag tiny localised defects; 5 to require a larger defect area.

Every parameter has a **(?) help icon** next to its label in the UI — hover for the in-app tooltip, which often contains more nuanced advice than this table.

Rule of thumb for your first model: dataset name → Wide ResNet50 backbone → image size 256 → defaults for the rest. Train, look at the test-bench results, and only change one parameter at a time if you need to tune.

7. Training runs in a background queue. The model list shows live progress in the *Training Activity* panel (active version with a progress bar). EfficientAD on a small dataset typically finishes in 1–5 minutes on a CUDA GPU, longer on CPU.

© 2026 Larix Technologies Kft.



8. When the version turns **Ready**, open the model detail page. Metrics (accuracy / precision / recall / F1) are shown for the new version.

   (Same model detail screenshot as above — version row shows Ready status and metrics once training completes.)

9. Click **Activate** on the version, then **Publish** at the top of the page. *Activate* marks this version as the one to use; *Publish* makes the model available for inspections.

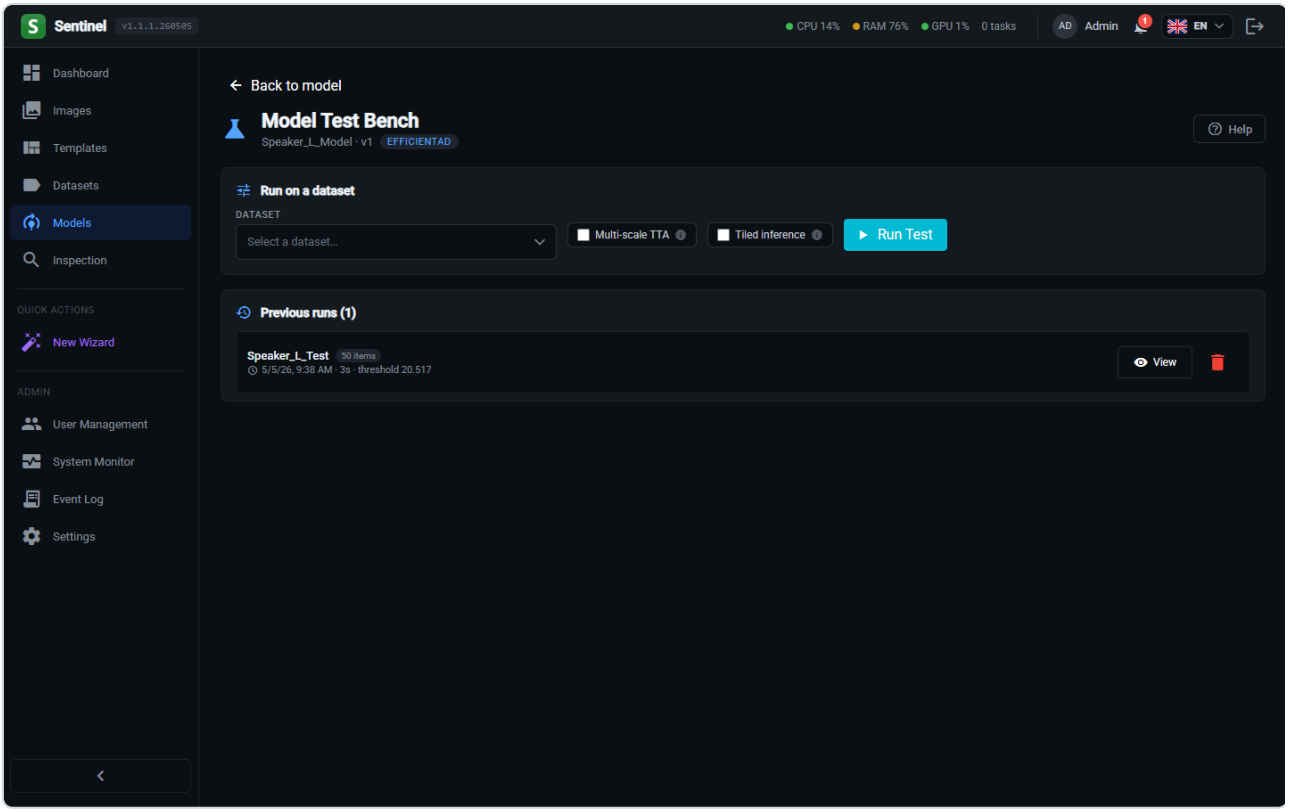
---

9. Testing the model on the bench

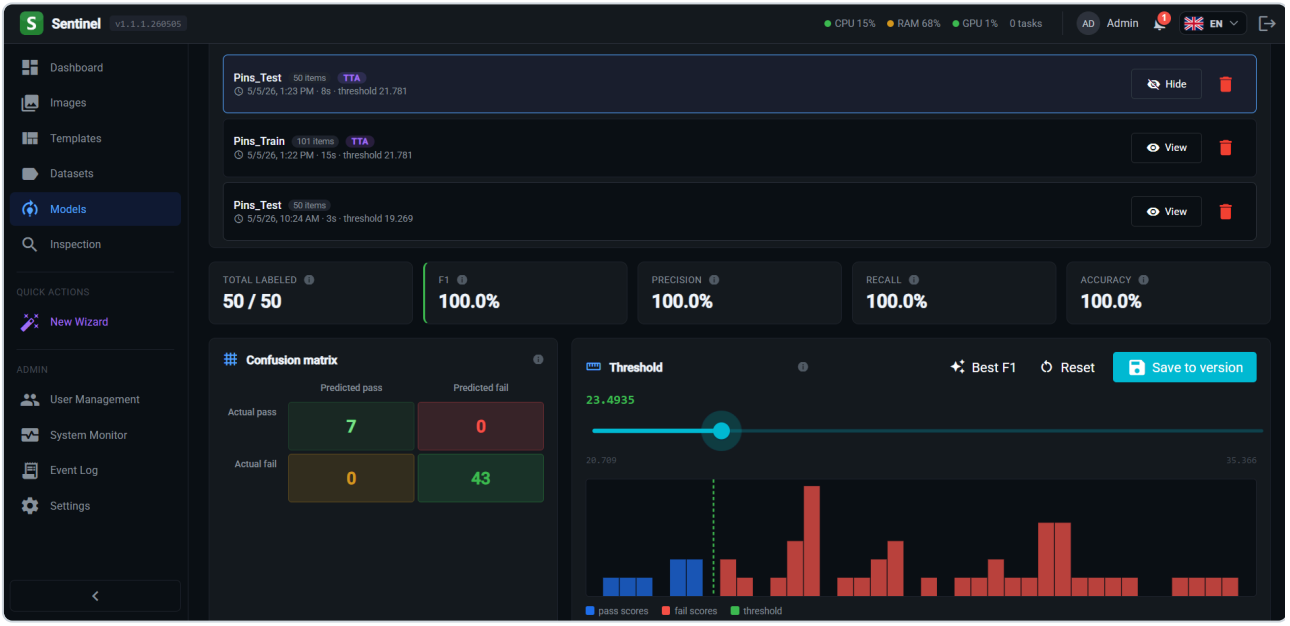
The **test bench** lets you inspect the model's predictions against your labelled dataset before running it on production images.

1. From the model detail page, on the active version row, click **Test bench**.
2. **Pick the *Test* dataset** in the dataset selector at the top — *not* the training dataset. Evaluating against the data the model trained on is misleadingly optimistic; the test bench's job is to show how the model handles unseen examples. Then click **Run**.
3. The bench loads the model and runs it against your dataset. The grid shows every image with its predicted vs. actual label, sorted with errors first by default.

© 2026 Larix Technologies Kft.



4. Click any thumbnail to open the **detail modal**. You see the original crop, the AI **heatmap** (red where the model thinks something is wrong), and the score.



5. **Navigate** with the chevron buttons or `←` / `→` arrow keys. Press `Esc` to close.
6. **Tune the threshold** if needed: drag the slider above the histogram to shift where Pass turns into Fail. Errors update live.
7. Use **Mark as FAIL** or **Exclude from training** in the modal if you spot a mislabelled crop. Then retrain.

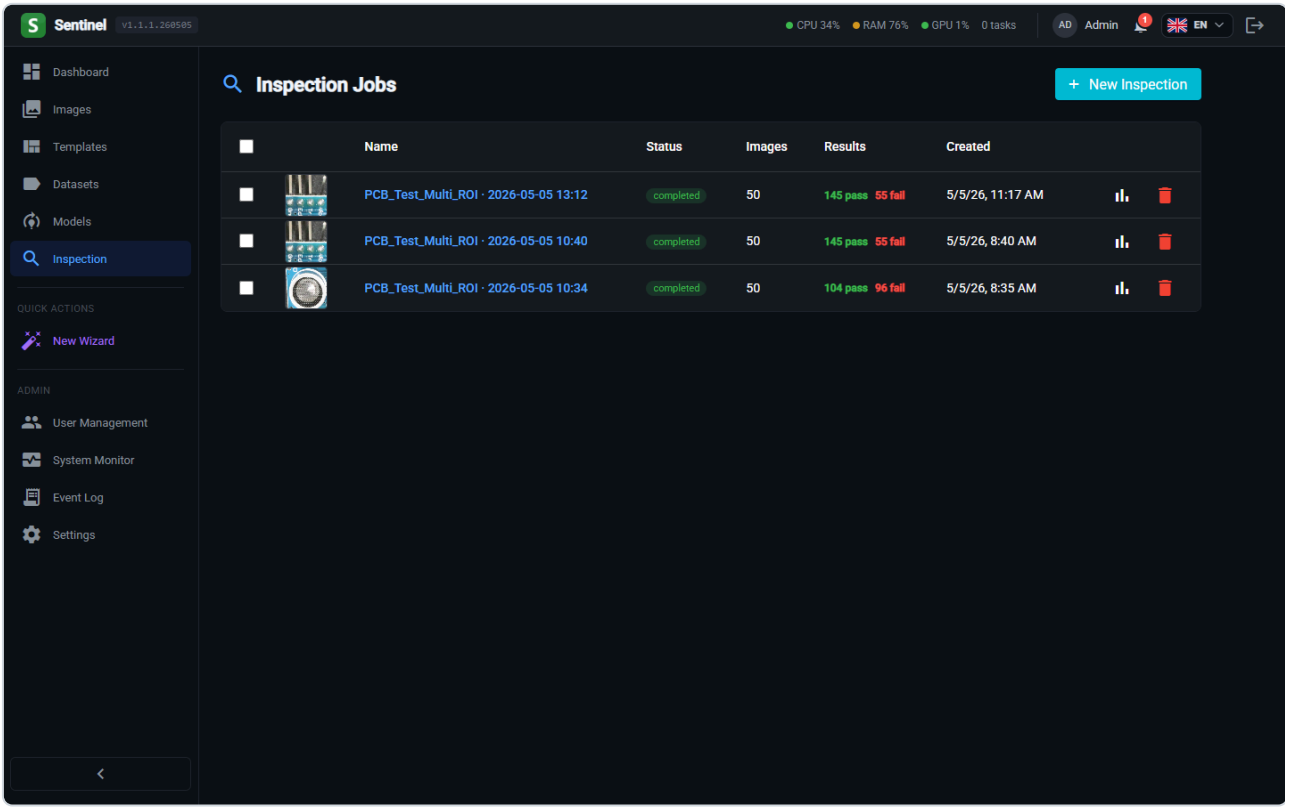
> If accuracy looks good and the heatmaps highlight the right areas, your model is ready. Otherwise, label more images, adjust the threshold, or retrain with different parameters.

© 2026 Larix Technologies Kft.

10. Running an inspection (optional)

The full workflow ends with running the model on a new batch of images.

1. Sidebar → **Inspection** → **New Inspection**.
2. Pick your template, assign your published model to the region, pick a batch of new images (e.g. images from *Sample - VisA PCB1 / Anomaly*).
3. Click **Start**. Results appear per image, per region, with pass/fail and confidence scores.



11. What next?

You now have a working end-to-end model. To use Sentinel on your own products:

- Photograph 100+ examples of your **good** product. Vary lighting and orientation slightly.
- Photograph at least 20–50 known **defective** examples for validation.
- Repeat the workflow above, replacing the *Sample - VisA PCB1* root with a folder of your own images.

For licensing, additional users, or feature requests, contact your Sentinel vendor.

© 2026 Larix Technologies Kft. — info@larix-technologies.com