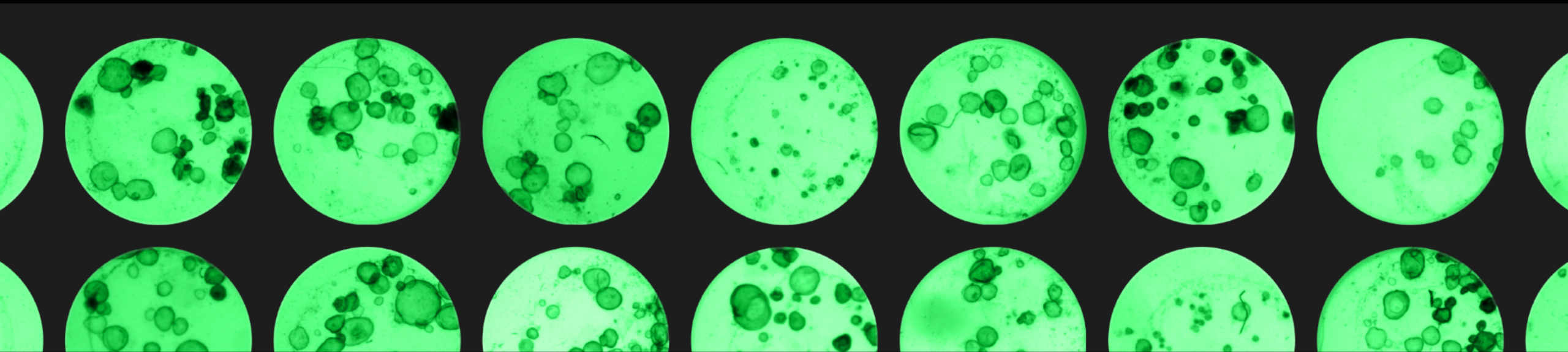
ramona

Is this model actually better?

How to judge object detection and segmentation models from training curves, evaluation metrics, and inference results.

MCAM Machine Learning · September 2026



Four questions this deck answers

What do the evaluation metrics mean?

IoU, Dice, precision, recall, mAP, count error, overlap ratio, harmonic mean. What each one measures and when it misleads.

Is my new model better than the old one?

Which numbers to compare, on which images, and how big a difference has to be before it counts.

Will it generalize?

The difference between a good test score and a model that works on next month's plates, and how to check.

How many images and labels do I need?

Software minimums, practical rules of thumb, and a method for finding out for your own assay.

Two model families in the MCAM software

	Object detection	Segmentation
Output	One bounding box per object	A class label for every pixel
Typical use	Organoid detection, counting, content-aware focus	Confluence, cell masks, region segmentation
Labels you draw	A box around every object in the image	Painted masks, which may cover only part of the image
Test metrics	IoU, Dice, count error, overlap ratio	Pixel IoU and Dice, per class plus harmonic mean
Watched during training	Precision, recall, mAP@0.5, mAP@0.5:0.95, losses	Training and validation loss and IoU
Typical epochs	200 to 1000	5 to 25
Starting weights	Organoid base weights	ImageNet EfficientNet-B3 encoder, cell-confluence base

Every training run follows the same path

Boxes or masks drawn in the software

→

3072 px images become 768 px samples, or native-resolution tiles

→

Train, validation, test. Split by camera image, so tiles never leak

→

Detection or segmentation, per-epoch history recorded

→

One portable inference file

→

6

Scored on the test set, logged to the model registry

The numbers you compare between models come from step 6, the held-out test set. Never from the images the model trained on.

Three subsets, three jobs

Train · 80%

The model learns from these images. Its score here is always flattering, because it has memorized them. Never compare models on training scores.

Validation · 12%

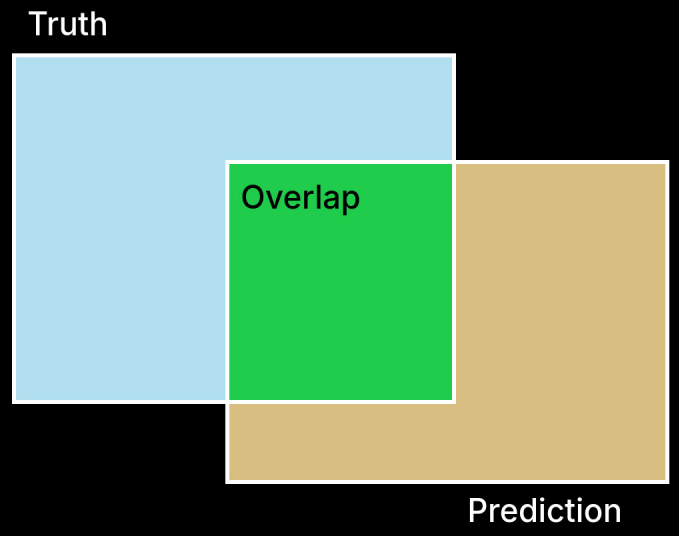
Scored after every epoch while training. Decides which checkpoint is kept: segmentation models save the epoch with the highest validation IoU.

Test · 8%

Never touched during training. Produces the IoU, Dice and count numbers written to the model registry. This is the scorecard.

A validation score is slightly optimistic too, because it chose the checkpoint. The test score is the honest one.

IoU: how much do prediction and truth overlap?



IoU = overlap area ÷ area covered by either

Dice = 2 × overlap area ÷ (truth area + prediction area)

Both range from 0 (no overlap) to 1 (identical). They measure the same thing on different scales, and Dice is always the larger number.

IoU	0.50	0.70	0.80	0.90
Dice	0.67	0.82	0.89	0.95

Compare IoU with IoU and Dice with Dice. A Dice of 0.82 is not better than an IoU of 0.75; it is worse.

Reading an IoU number

RULES OF THUMB FOR MCAM CELL AND ORGANOID WORK

Below 0.5

Not usable. Wrong objects, missed objects, or the labels and predictions disagree about what the task is.

0.5 to 0.7

Rough. Finds most things; edges drift. Fine for a first pass.

0.7 to 0.85

Good for counting and confluence.

Above 0.85

Excellent. Near the ceiling set by label noise.

Small and thin objects score lower

A one-pixel error on the edge of a 10 px cell is a big fraction of its area. The same visual quality on large organoids gives a higher IoU. Compare within a task, not across tasks.

The ceiling is human agreement

Two people labeling the same image rarely exceed 0.85 to 0.9 IoU with each other. A model cannot be scored higher than the labels are consistent.

Detection metrics: during training and on the test set

EVERY EPOCH, FROM THE TRAINER

Metric	What it answers
Precision	Of the boxes it drew, how many are real objects? Low precision means false alarms.
Recall	Of the real objects, how many did it find? Low recall means misses.
mAP@0.5	Precision and recall combined across confidence levels. A box counts as correct if it overlaps truth with IoU of at least 0.5.
mAP@0.5:0.95	The same, averaged over stricter overlap thresholds up to 0.95. Rewards tight boxes, not just roughly placed ones.

ONCE, ON THE HELD-OUT TEST SET

Metric	What it answers
IoU and Dice	Overlap between all predicted boxes and all true boxes, treated as masks. Overall placement quality.
Count error	Average percentage difference between predicted and true object count per image. The number that matters for counting assays.
Overlap ratio	Total predicted box area ÷ total true box area. 1.0 is ideal. Above 1 the model over-detects or draws oversized boxes; below 1 it misses or draws them small.

Segmentation metrics: pixels, not objects

Binary models

One IoU and one Dice over every labelled pixel in the test set. During training, validation IoU is computed each epoch and the best epoch is the one that is kept.

Multi-class models

IoU and Dice for each class separately, then a harmonic mean across the classes present in the test set. Report the per-class numbers, not only the mean.

Unlabelled pixels are scored nowhere

Only the pixels you painted count, in the loss and in every metric. You may label part of an image. What the model predicts elsewhere is neither right nor wrong.

What a segmentation score does not tell you

Pixel IoU says nothing about object count. Two touching cells merged into one blob can score 0.95 IoU and still give the wrong count. If the assay counts objects, check counts on overlays or use a detection model.

Why the harmonic mean for multiple classes

Class scores	Arithmetic mean	Harmonic mean	What you would conclude
cancer 0.85, t-cell 0.62	0.74	0.72	Both classes acceptable, t-cell weaker
cancer 0.90, t-cell 0.30	0.60	0.45	The arithmetic mean hides a failing class; the harmonic mean does not
cancer 0.90, t-cell 0.05	0.48	0.09	One class is essentially unrecognized

A minority class cannot hide

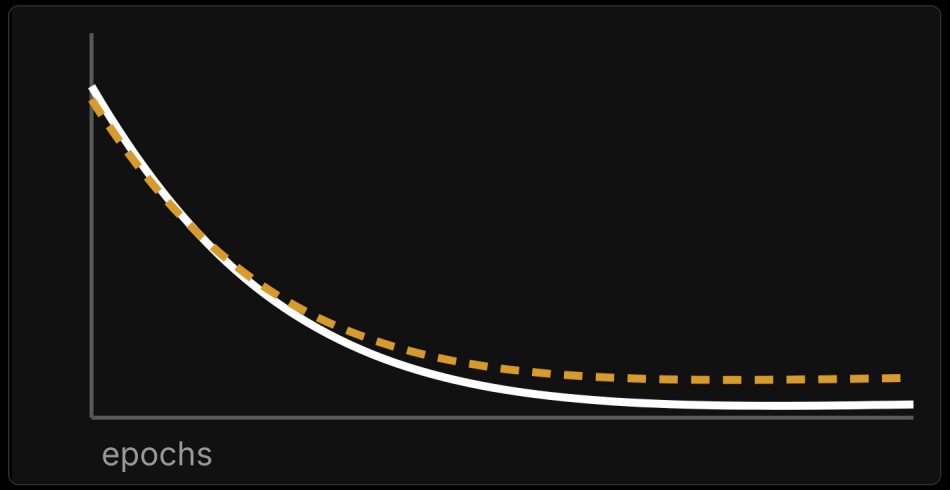
The harmonic mean is pulled toward the worst class. A model that is great on the abundant class and useless on the rare one gets a poor overall score, which is the right answer for most assays.

Absent classes are skipped

A class with no pixels in the test set would score 0 and drag the mean to 0. It is left out of the mean instead. Check that every class you care about is actually present in your test images.

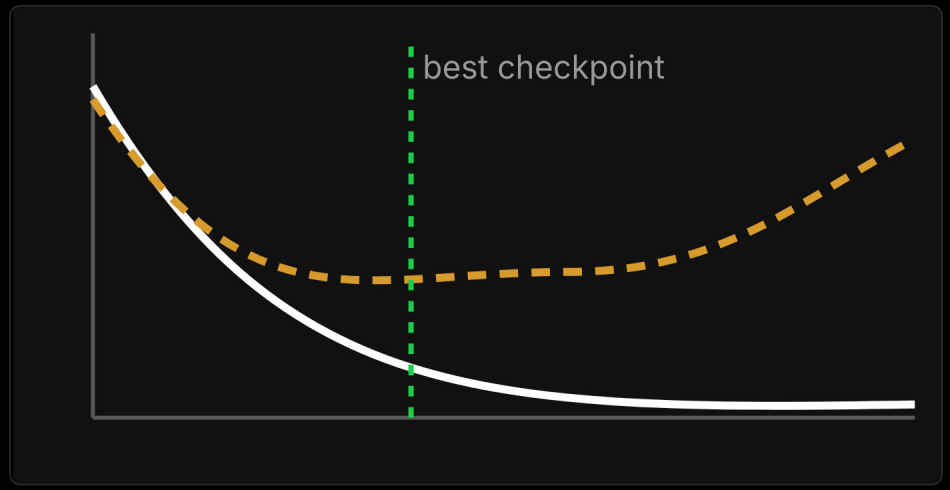
Rule: when two multi-class models are close on the harmonic mean, the per-class table decides.

Three shapes of a training run



Healthy

Both losses fall and flatten. Validation stays close to training. Validation IoU rises and plateaus. Stop here, or a bit later.



Overfitting

Training loss keeps falling while validation loss turns back up. The model is memorizing. The software keeps the best validation epoch, but the fix is more or more varied data.

training loss

validation loss

Underfitting: both losses stay high and flat. More epochs, higher resolution, a deeper encoder, or check the labels.

What the software records for every run

	Segmentation	Detection
Per epoch	Training and validation loss, training and validation IoU	Training loss, validation loss, precision, recall, mAP@0.5, mAP@0.5:0.95
Checkpoint kept	Epoch with the best validation IoU	Trainer's best weights
On the test set	IoU, Dice, per class if multi-class	IoU, Dice, count error, overlap ratio
Stored where	Model registry row, model_registry_entry.json	Same

EPOCHS ARE AUTO-SCALED BY SAMPLE COUNT

Labelled samples	Segmentation	Detection
under 100	25	1000
under 1,000	15	500
under 10,000	5	200
10,000 or more	1	200

Fewer samples get more passes over the data. Advanced mode lets you set epochs by hand.

Comparing two models fairly

1 · Same test images

Both models must be scored on the identical held-out set. A model trained later on a bigger dataset gets a different random split, so re-evaluate both on one fixed set.

2 · Same task and classes

Same subject, same class list, same labeling rules. Otherwise the numbers are not comparable at all.

3 · Same or noted resolution

Downsampled 768 px training and native-resolution tiling see different detail. Note which was used before reading the score.

4 · Test metrics only

Never training metrics. Validation metrics were used to pick the checkpoint, so prefer test.

5 · All the metrics

IoU and count error can disagree. For a counting assay, count error and overlap ratio win. For confluence, IoU wins.

6 · Look at the overlays

Numbers summarize. Open predictions on test images and on a fresh plate. Errors that matter to the assay are obvious by eye.

A worked comparison

ILLUSTRATIVE NUMBERS, SAME TEST SET OF ORGANOID IMAGES

Metric	Model A (last month)	Model B (retrained today)	Read
IoU	0.78	0.81	B places boxes slightly better
Dice	0.88	0.90	Same story as IoU, as expected
Count error	4%	9%	A counts more accurately
Overlap ratio	1.02	1.15	B over-detects or draws boxes too large
Test images	41	41	Small set: a 0.03 IoU gap is within noise

For a counting assay

Model A is better. Its count error is half of B's and its overlap ratio is close to 1. The higher IoU of B does not help the assay.

For a focus or crop task

Model B is marginally better placed, but 0.03 IoU on 41 images is not decisive. Retrain with the same split or pool more test images before switching.

A small difference may be noise

With 8% of images held out, a 50-image dataset leaves 4 test images. Which 4 landed there decides the score as much as the model does.

- On small test sets, treat IoU differences under about 0.03 as a tie.
- Retrain each model twice with different random splits. If the winner changes, there is no winner.
- Prefer a model that wins on several metrics and by eye, not one that wins one metric by a hair.
- Keep a fixed, shared test set for a task so every model over time is scored on the same images.

Dataset size	Test images at 8%
25	2
50	4
100	8
500	40
2,000	160

Below roughly 20 test images, a single mislabeled image moves the score by several points.

How do I know a model generalizes?

What the test score proves

Unseen images from the same experiments it trained on: same plates, same day, same illumination, same cell line. It answers: did it memorize or did it learn?

What it does not prove

That it works on next month's plate. Images from one plate are correlated. A held-out image from the same plate is much easier than an image from a new one.

The real test

Hold out an entire experiment, ideally a different plate, day and operator. Never add it to training. Run inference and compare counts or confluence with what you expect by eye.

Where models usually break

Exposure and illumination changes · different magnification or binning · much higher or lower cell density than in training · debris, bubbles and plate edges · a new plate type or well geometry · a different fluorescence channel or channel order

If the model degrades on the held-out experiment, the fix is almost always to add labeled images from that kind of condition, not more epochs.

How many images and labels do I need?

Stage	Labeled images	Labeled objects or painted area	What to expect
Smoke test	3 to 10	A few dozen objects	Trains (the software requires 3). Proves the pipeline, not the model.
Usable prototype	20 to 50	Several hundred objects, spread across wells	Decent on the same conditions. Expect misses on anything unusual.
Production	100 to 500 or more	Thousands of objects, from several plates and days	Stable scores, generalizes to new plates of the same assay.

Count objects, not images

One image with 300 organoids teaches more than ten images with 5 each. The registry's epoch scaling is keyed to labelled samples for this reason.

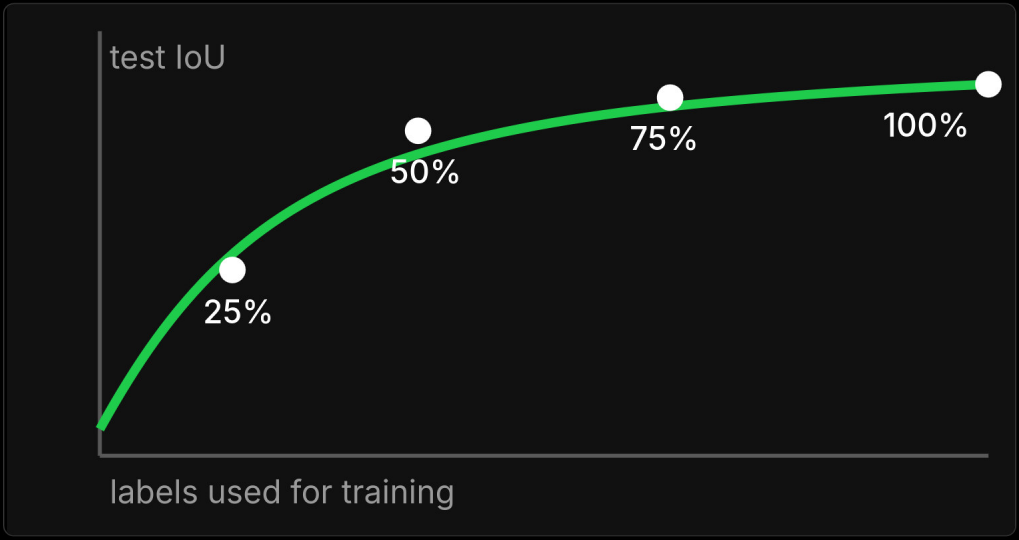
Diversity beats volume

Spread labels across plates, days, densities and edge cases. A model trained on one plate learns that plate.

Starting weights shrink the need

Detection starts from organoid base weights and segmentation from a confluence base, so a related task needs far fewer labels than training from scratch.

The honest answer: add data until the score stops moving



A learning curve for your own assay

1. Fix one test set and keep it out of everything.
2. Train on a quarter, half, three quarters and all of the remaining labels.
3. Plot test IoU (or count error) against label count.
4. Still climbing at 100%: label more of the same kind.
5. Flat: more labels of this kind will not help. Spend effort on diversity, label quality, or resolution instead.

Label quality beats label quantity

DETECTION BOXES

- **Box every object in the image.** An unlabeled object teaches the model that it is background, which directly lowers recall.
- Keep box tightness consistent. Loose boxes in some images and tight ones in others cap mAP@0.5:0.95.
- Decide a rule for partial objects at the well edge and touching objects, then apply it everywhere.

SEGMENTATION MASKS

- **Partial labels are fine.** Unpainted pixels are ignored, so paint fewer regions accurately rather than whole images sloppily.
- Paint some background too, especially near edges and debris, or the model never learns what 'not cell' looks like there.
- Label the hard regions: touching cells, boundaries, dim objects. Easy centers add little.

Two labelers, one rule book. Inconsistent labels set a ceiling on every metric that no amount of training can pass.

Signals at inference time

Confidence threshold (detection)

Raising it trades recall for precision. Two models compared at different thresholds are not comparable. Pick the threshold on validation images, then freeze it.

Resolution and context

Downsampling a 3072 px image to 768 px loses small objects; tiling keeps them but the default segmentation model sees only about 111 px of context per pixel. Context-defined classes such as a smooth organoid center need a deeper encoder or downsampling.

Sanity checks on fresh plates

Do counts track what you see by eye? Does confluence move the way the experiment should? A model that scores well but disagrees with the biology is wrong somewhere.

Speed is a metric too

Segmentation in float16 runs roughly 1.5 to 1.8× faster on GPU for the same accuracy. If a model is only slightly better but twice as slow, it may not be the better choice for a live workflow.

Mistakes that make a worse model look better

Scoring on training images

Any model looks great on images it has seen. Use the test set.

Different test sets

Two registry rows from two splits are not comparable. Re-evaluate on one shared set.

Reading one number

High IoU with high count error, or a great harmonic mean with one collapsed class. Read the whole row.

Test images that leak

Tiles or near-duplicate frames of the same field in both train and test. The pipeline splits by camera image to prevent this; do not merge datasets that re-image the same wells.

Comparing at different thresholds or resolutions

Confidence threshold, tile versus downsample, and encoder depth all change the score without changing the model's real quality.

Trusting a big win on a tiny test set

Four test images cannot separate two models. Widen the test set before deciding.

Deciding: is model B better than model A?

1. Both evaluated on the same fixed test set, same classes, same resolution and threshold.
2. Training curves for B look healthy: validation tracks training, no late rise in validation loss.
3. B wins on the metric the assay consumes: count error for counting, IoU for confluence and masks.
4. B does not lose badly on any other metric or any single class.
5. The margin is bigger than the noise for that test-set size.
6. Overlays on a held-out experiment from a different plate or day look at least as good.

All six: switch to B and record why in the registry notes. Any miss: B is not better yet.

Glossary

Term	Meaning
Epoch	One full pass of the training set through the model.
Loss	The number the optimizer drives down during training. Lower is better; its absolute value is not comparable between model types.
IoU / Dice	Overlap between prediction and truth. Dice is always the larger of the two for the same prediction.
Precision / Recall	False-alarm rate versus miss rate for detections.
mAP	Mean average precision: precision and recall combined over all confidence levels, at one or several IoU thresholds.
Count error / overlap ratio	Detection test metrics for object count accuracy and total box area relative to truth.
Harmonic mean	A per-class average that is dragged toward the worst class.
Checkpoint	Saved model weights from one epoch. The best validation epoch is the one kept.
Overfitting	Training score improves while validation gets worse: the model is memorizing.
Domain shift	Inference images differ from training images in illumination, density, plate, or channel.