Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.083365

ARTICLE

# From Public Benchmarks to a Low-Resource Target Domain: A Comparative Study of Wood Surface Defect Detection

**Khanh Nguyen-Trong**[1,*] **and Tan Nguyen-Thi-Thanh**[2]

[1]Intelligent Computing for Sustainable Development Lab, Faculty of Information Technology, Posts and Telecommunications Institute of Technology (PTIT), Hanoi, Vietnam

[2]Faculty of Information Technology, Electric Power University, Hanoi, Vietnam

*Corresponding Author: Khanh Nguyen-Trong. Email: khanhnt@ptit.edu.vn

Received: 02 April 2026; Accepted: 01 June 2026; Published: 23 July 2026

**ABSTRACT:** Automated wood surface defect detection is difficult to evaluate reliably because defects are often small, low-contrast, and visually confounded by natural wood texture, while reported performance can vary substantially with benchmark design and domain shift. To address this issue, we conduct a comparative study across three practically relevant settings: a curated seven-class benchmark, a broader in-domain seven-class protocol derived from the same source dataset, and supervised adaptation to a low-resource Vietnamese target domain. We compare lightweight two-stage detectors based on Faster Region-based Convolutional Neural Network (Faster R-CNN) with MobileNetV3-FPN against a compact You Only Look Once version 8 (YOLOv8s) baseline, while also testing two small-object-oriented YOLO refinements as targeted diagnostic variants rather than as the primary claimed contribution. Across in-domain experiments, the compact YOLOv8s baseline delivers the strongest performance, achieving 84.38% AP50 on the curated benchmark, whereas performance drops to 81.16% AP50 under the broader protocol, indicating that benchmark breadth materially changes the apparent difficulty of the task and the relative strength of competing models. In the target-domain setting, source-initialized fine-tuning improves optimization behavior and can outperform target-only training in a representative single run, but repeated-seed evaluation does not confirm a stable held-out-test advantage under the same adaptation budget. These findings suggest that conclusions drawn from a single curated benchmark may overstate model robustness, and that for wood defect detection, protocol breadth and source-to-target shift should be treated as central evaluation factors rather than secondary experimental details.

**KEYWORDS:** Wood surface defect detection; lightweight detector comparison; MobileNetV3-FPN; YOLOv8; benchmark sensitivity; industrial visual inspection; domain shift

## 1 Introduction

Automatic surface inspection is important in industrial production because manual inspection is costly, subjective, and difficult to scale. In wood processing, the problem is particularly challenging because natural grain, resin, knots, cracks, pith, and staining introduce substantial appearance variation even before true defects are considered. Surface defects are therefore relevant not only to visual grading but also to downstream material value and utilization [1].

In practice, an inspection system must therefore detect visually weak defects, separate semantically similar subtypes, and remain stable across production batches while using detector choices that are realistic for practical inspection rather than only optimal on a curated benchmark.

From a vision perspective, wood surface defect detection is challenging because many targets are small, low-contrast, elongated, or only partially visible, several categories such as live knots, dead knots, knots with cracks, and cracks are semantically close, the class distribution is long-tailed, and deployment data often differ from benchmark data in scale, framing, and label space.

Recent wood-defect studies mainly pursue detector-side improvements, typically by modifying feature fusion, attention, small-target handling, or localization loss [2–6]. However, it remains unclear whether the resulting detector rankings remain stable when the evaluation protocol is broadened or when models are transferred to a lower-resource target domain. Across these papers, the same trade-offs keep resurfacing: small targets disappear at coarse scales, localization is fragile on weak or elongated defects, and the detector still has to fit a practical compute budget.

Beyond curated benchmark performance, the central unresolved issue is whether detector rankings remain stable when evaluation protocols are broadened and when models are transferred to a lower-resource target domain. This question is practically relevant in settings where labeled wood-defect data are limited, including emerging datasets such as VNWoodKnot. We are therefore interested not only in which detector performs best on the curated benchmark, but also in how well conclusions drawn from that source setting survive broader evaluation and source-to-target transfer. In practical terms, this study asks three questions. Do strong results on the curated benchmark still hold when the evaluation protocol is broadened? Does detector family matter more than a small module-level refinement? And when the model is moved to a different dataset, how much performance is lost as label space, framing, and object scale change?

Given practical compute constraints, we focus on lightweight models: Faster R-CNN with MobileNetV3-FPN, a compact YOLOv8s baseline, and two small-object-oriented YOLO variants with a P2 head and a custom WNIoU-style hybrid box loss. We study them on two datasets: the VSB dataset [1] and VNWoodKnot, a smaller low-resource target dataset collected in Vietnam [7]. Rather than presenting a new detector module as the central contribution, this paper is designed as a comparative evaluation of detector-family choice, protocol breadth, and source-to-target transfer.

This study makes the following contributions.

1. We present a comparative evaluation spanning a curated benchmark (VSB), a broader in-domain seven-class protocol, and a low-resource target-domain setting.
2. We compare lightweight two-stage detectors and a compact one-stage detector family under shared processing and evaluation conditions, while testing two targeted YOLO refinements as diagnostic small-object-oriented variants.
3. We show that benchmark breadth materially affects apparent detector performance under the studied protocols.
4. We analyze both single-run and repeated-seed transfer behavior on VNWoodKnot under a fixed fine-tuning budget.

## 2 Related Work

Recent works on wood defect detection commonly evaluate YOLO-style detectors on benchmark variants derived from the VSB dataset, including ODCA-YOLO, BPN-YOLO, SiM-YOLO, FDD-YOLO, and CWB-YOLOv8 [2–6]. Although these models differ in where they add capacity, the overall recipe is similar: they retain a one-stage backbone and then introduce feature fusion, attention, small-target handling, or a modified localization loss to improve the reported score. The large-scale image dataset of wood surface defects introduced by Kodytek et al., which we refer to here as the VSB dataset, has become the common

reference point for this line of work because it offers a clearer taxonomy and a more systematic benchmark than many earlier in-house collections [1].

Wood inspection research is broader than bounding-box detection, however. Kilic et al. [8,9] study anomaly-oriented and hybrid sensing settings, and Hoang et al. [10] report 60.6% AP and 81.2% AP50 for an area-attention detector on a relabeled benchmark variant built from the Kodytek dataset, but with a relabeling scheme different from the original VSB taxonomy. Their method also exceeds the YOLOv12 reference reported in the same paper, which reaches 55.3% AP and 74.3% AP50. This is a useful neighboring reference because it is built from the same image source but uses a different label definition and evaluation framing. More broadly, it shows that wood inspection is still evaluated under multiple benchmark variants rather than under one universally accepted setup. Even within detector papers, the result format is usually the same: one curated split, one headline AP50, and one proposed architectural change. That convention makes the literature easy to scan, but it leaves open a practical question that matters to deployment: is the reported gain attached to the detector itself, or to the way the benchmark was screened and measured?

The broader industrial-inspection literature looks at the same problem from a slightly different angle. Reviews of edge-oriented and industrial defect detection put latency, memory footprint, and backbone efficiency on the same footing as accuracy [11]. That is one reason two-stage detectors such as Faster R-CNN remain useful as controlled references, especially when paired with lightweight backbones such as MobileNetV3 [12,13]. At the same time, YOLO-style one-stage detectors have become the practical default in many inspection tasks because they offer dense prediction with a favorable speed–accuracy trade-off.

The same trade-off shows up in compact detectors for PCB, aluminum, printed-surface, metallic, rail, and steel inspection, where authors repeatedly return to the same toolbox: lighter backbones, efficient heads, multiscale fusion, and revised box losses [14–19]. Small-object surveys tell a similar story. High-resolution heads and alternative localization losses are standard responses when targets are small, weakly contrasted, or easy to miss after repeated downsampling [20,21]. That background is exactly why our study keeps both ingredients in view: detector-family choice and small-object-oriented YOLO refinements.

Cross-paper detector results are still hard to compare cleanly. Reviews and benchmarking papers keep making the same point: datasets, split rules, and metrics are not reported in a fully consistent way, so cross-paper ranking can look firmer than it really is [5]. In wood inspection, that matters immediately. Changing class filters, screening rules, or sampling policy can change which model looks best. A detector that performs well on a small curated subset may not keep the same advantage once the protocol is broadened, and neither result says much by itself about transfer to a different dataset. That is the point of departure for our study. We keep the detector set compact and ask two practical follow-up questions: what still holds when the source-domain protocol is broadened, and what still holds when the model is moved to VNWoodKnot?

## 3 Materials and Methods

### *3.1 Datasets and Protocol Design*

In this study, we use the VSB dataset as the source dataset for both the curated benchmark and the broader seven-class protocol, whereas VNWoodKnot is used as the low-resource target domain for adaptation experiments. The in-domain experiments are built on the large-scale wood surface defect dataset of Kodytek et al. [1]. Similarly to previous studies on this dataset, we derive a seven-class label space consisting of live_knot, dead_knot, resin, knot_with_crack, crack, marrow, and knot_missing. This shared label space is used for both protocols so that protocol sensitivity can be analyzed without changing the task definition.

To reflect that Vietnamese local setting, we use VNWoodKnot [7], a recent dataset collected in Vietnam for wood knot detection and classification. It consists of board-surface images acquired under a fixed local

imaging setup and annotated around knot-related categories, making it a useful low-resource target domain for this study. The original VNWoodKnot label space contains knot_free, live_knot, and dead_knot. In our study, the two knot classes form the detection task, while knot_free images are preserved as negative samples. This makes VNWoodKnot a practically relevant low-resource target domain for evaluating supervised adaptation rather than a direct seven-class benchmark.

In the standard `T0`/`T1` pipeline, VNWoodKnot images are not resized offline to 1024 in advance; resizing to the training resolution is performed on the fly during training. The knot_free images are retained as background-only negative samples rather than dropped, since removing them would discard a substantial fraction of the available target-domain data and bias training toward defect-bearing images. Under the reported split, negatives account for roughly one third of the images in each of train, validation, and test, which preserves a consistent negative-to-positive ratio across splits. No hard-negative mining and no oversampling are applied, so the contribution of negatives is determined by their natural proportion in each batch rather than by a sampling heuristic.

Fig. 1 gives a visual comparison, while Table 1 summarizes the main quantitative differences between the two datasets. These differences matter because the VSB source dataset mainly tests small-defect localization under a richer seven-class taxonomy, whereas the target domain emphasizes larger knots under a much smaller label space.


**a** The VSB source dataset. **b** Representative images from VNWoodKnot.

**Figure 1:** Visual comparison of the two datasets used in this study.

**Table 1:** Comparison between the VSB source dataset and the low-resource target domain used in this study.

| Attribute | Curated Benchmark (VSB Source) | Low-Resource Target Domain (VNWoodKnot) |
|---|---|---|
| Images and annotations | 20,276 and 43,974 | 1515 and 1021 |
| Label space used in this study | 7 classes: live_knot, dead_knot, resin, knot_with_crack, crack, marrow, knot_missing | 2 classes (live_knot, dead_knot) with knot_free retained as negatives |
| Image format | Heterogeneous source images converted into tiled training/evaluation records | Fixed-size images (1500 × 1500) |

(Continued)

**Table 1 (continued)**

| Attribute | Curated Benchmark (VSB Source) | Low-Resource Target Domain (VNWoodKnot) |
|---|---|---|
| Derived tiles/processed images | 48,156 tiles after preprocessing (38,493 train, 4879 val, 4784 test) | No tiling used in the reported target-domain setting |
| Reported split sizes (train/val/test) | 38,493/4879/4784 tiles | 1060/226/229 images and 715/151/155 boxes |
| Background-only images | Not emphasized in the benchmark protocol | 500 knot_free images |
| Main implication | Strong emphasis on small-defect localization and long-tail multi-class discrimination | Stronger emphasis on larger knot regions, simpler label space, and source-to-target transfer |

The mean normalized bounding-box area in the in-domain dataset is 0.008857, whereas VNWoodKnot has a mean normalized bounding-box area of 0.119985. This order-of-magnitude gap indicates a strong difference in object scale and framing. Together with the label-space reduction and the presence of many negative images in VNWoodKnot, these statistics support interpreting the external evaluation as a substantial source-to-target shift rather than as a minor threshold-calibration issue.

### *3.2 Protocol Variants*

#### *3.2.1 Protocol A: Curated Seven-Class Benchmark*

We define the curated benchmark to be comparable in spirit to prior seven-class benchmark variants derived from the Kodytek dataset [1]. Following prior studies [2–6], we first select 3600 source images and assign train, validation, and test splits at the source-image level before generating tiled records for detection training and evaluation. All tiles derived from the same source image therefore remain in the same split. This benchmark contains 7679 training tiles, 977 validation tiles, and 972 test tiles. For experiments on this protocol, we retain short-budget 50-epoch runs for detector screening and ablation, and we also include a 200-epoch run as an extended follow-up schedule.

#### *3.2.2 Protocol B: Broader Seven-Class Protocol*

To check whether conclusions from the curated benchmark remain valid under the broader seven-class protocol, we also evaluate a broader seven-class protocol constructed by filtering the full dataset down to the same seven classes. This protocol produces a larger validation split with 4846 validation tiles under the present tiling configuration. As in Protocol A, we keep the original 50-epoch run as the short-budget reference, and we also include a 200-epoch YOLO run to better match the longer schedules commonly used in prior YOLO-based studies.

#### *3.2.3 External Protocol: VNWoodKnot Adaptation Setting*

For the external study, VNWoodKnot is reformulated as a two-class detection task containing `live_knot` and `dead_knot`, while `knot_free` images are preserved as negatives. This setting allows us to measure how quickly a curated-benchmark checkpoint can adapt to a target wood-inspection dataset under a fixed fine-tuning budget.

### *3.3 Detector Variants*

Based on the defect characteristics discussed in Sections 1 and 3.1, we select three detector families to cover complementary operating points in a practical wood-inspection design space. First, Faster R-CNN with MobileNetV3-FPN provides a proposal-based two-stage family that combines explicit region refinement with lightweight backbones, making it a relevant reference when localization control and computational efficiency must be balanced [11–13]. Second, YOLOv8s represents a compact one-stage family that has become a strong practical default in recent defect-inspection and wood-defect studies because of its favorable speed–accuracy trade-off and multiscale dense prediction behavior [2–5]. Third, we test two targeted YOLO refinements, namely a higher-resolution P2 head and a custom WNIoU-style hybrid box loss. The latter combines a CIoU term, an NWD-like similarity term, and a Wise-IoU-style focusing factor to test whether more robust localization supervision helps thin cracks and irregular knot-related regions [22–24]. This detector selection allows the study to address two practical questions at once: which detector family is most suitable for the present wood-defect setting, and whether widely motivated small-object-oriented refinements materially change that ranking.

#### *3.3.1 Low-Capacity MobileNet Baseline*

The lowest-capacity reference in our study is a Faster R-CNN detector instantiated from Faster R-CNN + MobileNetV3-320-FPN, with the image size fixed to $1024 \times 1024$. We include this model as a lower-cost anchor point for the proposal-based family, so that the study can distinguish between the effect of detector family and the effect of simply scaling backbone capacity inside the same two-stage design [12,13].

#### *3.3.2 High-Resolution (HR) MobileNet Baseline*

The main lightweight family in the paper uses Faster R-CNN + MobileNetV3-Large-FPN, trained at $1024 \times 1024$. This family appears in one broader-protocol run and several curated-benchmark variants summarized later in the experiment tables. We use it as the principal lightweight two-stage reference because it retains the proposal-based refinement behavior of Faster R-CNN while providing a stronger MobileNet backbone than the lower-capacity baseline above [12,13].

#### *3.3.3 YOLOv8s Baseline (Y0)*

The YOLOv8s runs on the broader seven-class protocol and on the curated benchmark use the standard YOLOv8s architecture. We choose this network as the compact one-stage baseline because it provides a modern, strong, and deployment-relevant detector family that is close in spirit to the YOLO-style models dominating recent wood-defect literature [2–5]. The final reported run on the curated benchmark is denoted `Y0-3600e200`.

#### *3.3.4 P2-Head Ablation (Y1)*

`Y1` adds an extra P2 detection head to YOLOv8s via a modified model configuration. This variant is motivated by a task-specific hypothesis: several wood defects in the present benchmark occupy small image regions or exhibit weak local contrast, so discriminative cues may be attenuated when prediction begins only from coarser pyramid levels. In feature-pyramid notation, the baseline YOLOv8s branch predicts from $\{P_3, P_4, P_5\}$, whereas `Y1` extends this set to $\{P_2, P_3, P_4, P_5\}$. Adding a higher-resolution detection layer therefore tests whether earlier multiscale prediction improves the representation of small defects in this domain, as suggested by recent small-object detection studies [20,21]. In the present study, `Y1` is

examined first as a 50-epoch screening-stage ablation and then as a matched 200-epoch comparison on the curated benchmark.

### *3.3.5 WNIoU-Style Box-Loss Ablation (Y2)*

`Y2` modifies the box regression loss to a custom WNIoU-style hybrid loss. In our implementation, this loss combines a CIoU term, an NWD-like similarity term, and a Wise-IoU-style focusing factor, rather than reproducing a single published formula verbatim [22–24]. This variant addresses a different hypothesis from `Y1`: some wood defects, particularly thin cracks and irregular knot-related regions, may be limited less by feature resolution than by unstable geometric supervision during box regression. A distance-aware and reweighted localization loss is therefore tested as a compact way to improve localization robustness without modifying the detector architecture itself. In the comparison reported here, `Y2` appears as a matched 200-epoch curated-benchmark rerun under the same long schedule as the final compact baseline.

For the YOLO branch, we write the detection objective as

$$\mathcal{L}_{\text{det}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{dfl}} + \mathcal{L}_{\text{box}}. \tag{1}$$

The baseline YOLOv8s and the P2-head variant keep the standard CIoU-based box term,

$$\text{CIoU} = \text{IoU} - \frac{\rho^2(\mathbf{b}, \mathbf{b}^*)}{c^2} - \alpha_{\text{ciou}} v \tag{2}$$

where $\mathbf{b}$ and $\mathbf{b}^*$ are the predicted and target boxes, $\rho^2(\cdot,\cdot)$ is the squared center distance, $c$ is the diagonal length of the smallest enclosing box, and $v$ is the aspect-ratio consistency term [24]. For `Y2`, we replace $\mathcal{L}_{\text{box}}$ with a hybrid form. First, an NWD-style similarity is computed on normalized boxes,

$$S_{\text{nwd}} = \exp\left(-\frac{\sqrt{(c_x - c_x^*)^2 + (c_y - c_y^*)^2 + \frac{1}{4}\left[(w - w^*)^2 + (h - h^*)^2\right]}}{s}\right), \tag{3}$$

where $(c_x, c_y, w, h)$ and $(c_x^*, c_y^*, w^*, h^*)$ denote predicted and target box centers and sizes, and $s$ is a distance scale (set to 0.05 in this study). We then form the blended base loss

$$\mathcal{L}_{\text{base}} = (1 - \lambda)(1 - \text{CIoU}) + \lambda(1 - S_{\text{nwd}}), \tag{4}$$

with $\lambda = 0.30$, and apply a Wise-IoU-style difficulty weight

$$w_{\text{focus}} = 1 + \beta(1 - \text{CIoU})^{\gamma}, \tag{5}$$

with $\beta = 0.50$ and $\gamma = 1.00$. The final custom box term is therefore

$$\mathcal{L}_{\text{box}}^{\text{Y2}} = w_{\text{focus}}\, \mathcal{L}_{\text{base}}. \tag{6}$$

This design preserves the original YOLOv8 classification and distribution-focal terms while changing only the localization supervision.

## 4 Experiments and Results

### *4.1 Training Setup and Evaluation Settings*

#### *4.1.1 Hyperparameters and Metrics*

We use a shared YOLO training setup wherever possible, including image size 1024, mixed-precision training, AdamW as the optimizer, and training seed 42. The experiments are organized in two stages. First, we use a 50-epoch budget for detector-family comparison and short-budget ablation analysis inside the study. Second, we extend the curated-benchmark YOLO branch to 200 epochs to test whether the short-budget ranking remains stable under a longer schedule.

The 50-epoch schedule is used as a practical short-budget setting for detector-family comparison and local ablation. It was chosen because preliminary runs show all tested detectors reach a stable validation regime well before epoch 50—consistent with Fig. 2, where AP50 peaks at epoch 55 and plateaus afterward—while still being short enough to support repeated-seed and ablation experiments under our compute budget. The 200-epoch schedule is used as an extended follow-up to check whether the short-budget ranking remains stable under longer convergence, and it aligns with training horizons commonly reported in recent YOLO-based wood-defect studies [2–6], allowing the final curated-benchmark numbers to be compared with prior work on a comparable basis.

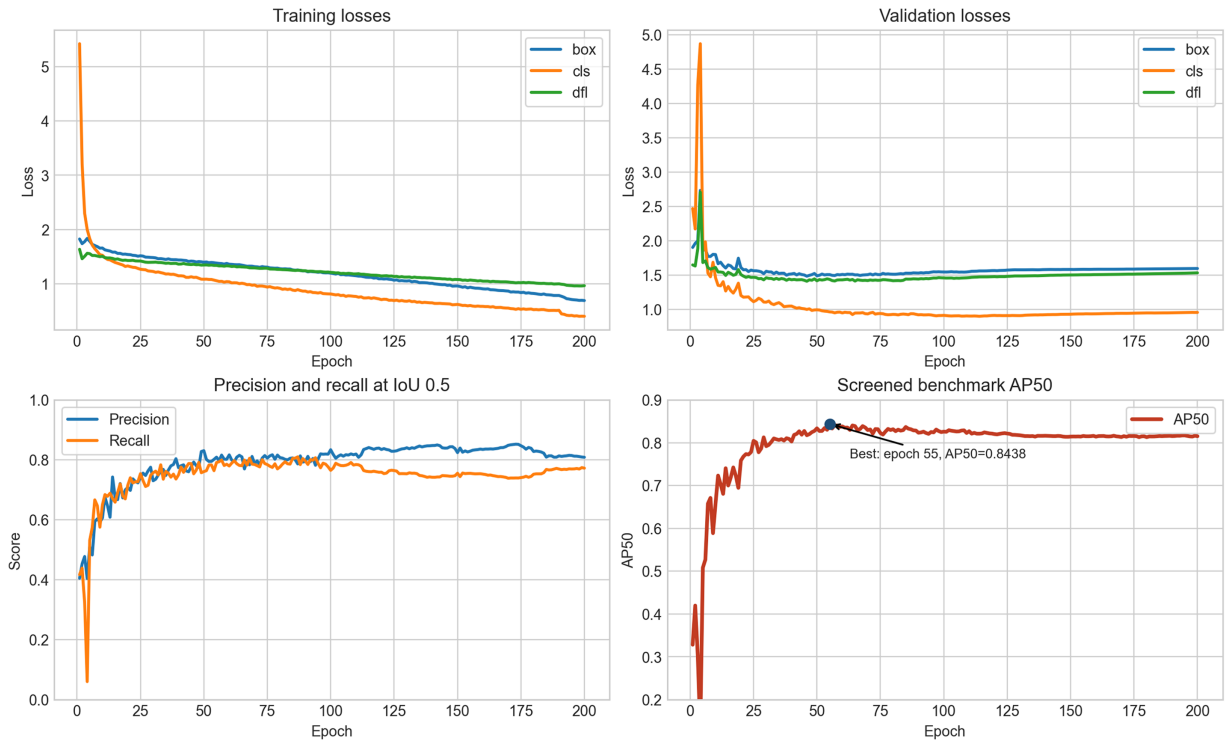


**Figure 2:** Training progress of the 200-epoch YOLOv8s run on the curated benchmark. The model reaches its best AP50 at epoch 55 and then stays in a stable high-performance regime.

All models are trained and evaluated under a consistent export and evaluation pipeline. For VNWood-Knot adaptation, both runs (`T0` and `T1`) use the same two-class target-domain split and are evaluated on the held-out target-domain test set with the same evaluator.

To stay comparable with recent YOLO-based wood-defect studies, we use AP50 as the primary metric. We also report precision and recall at IoU 0.5 (denoted Prec.50 and Rec.50).

In addition, we monitor a small-target subset metric for in-domain evaluation. Under the rule used in the dataset audit and evaluator, an annotation is treated as small if its normalized bounding-box area is ≤0.01, or its width is ≤16 pixels, or its height is ≤16 pixels, with `combine=any`. These auxiliary values are reported for completeness but are not emphasized in the main result tables.

For VNWoodKnot, the external comparison is a supervised two-class target-domain evaluation. The two classes are `live_knot` and `dead_knot`, while `knot_free` images are preserved as negative samples. This makes the external study a matched adaptation problem rather than an overlap-only label-mapping exercise.

### *4.1.2 Experiment Scenarios*

We evaluate four scenarios.

- Detector-family exploration (`B1`, `E1`): We compare lightweight Faster R-CNN + MobileNet baselines and compact YOLO baselines to identify a practical detector family before testing local YOLO refinements.
- Curated benchmark (`Y0-3600`, `Y0-3600e200`, `Y1`, `Y1-e200`, `Y2-e200`): Train and evaluate on the curated benchmark using `Y0-3600` as the short-budget reference and `Y0-3600e200` as the final literature-facing confirmation run. The P2 variant is reported both as a 50-epoch screening-stage comparison and as a matched 200-epoch rerun, whereas the WNIoU-style loss variant is included as a matched 200-epoch rerun under the same curated-benchmark protocol.
- Broader seven-class protocol (`Y0-full`, `Y0-full_e200`): Train and evaluate on the broader seven-class protocol to quantify the protocol-sensitivity gap relative to the curated benchmark. We retain the original 50-epoch checkpoint as the stronger study-internal reference, and we use an additional 200-epoch rerun as a diagnostic check on schedule sensitivity.
- Target domain adaptation (`T0`, `T1`): Compare supervised target-only training (`T0`) and curated-benchmark source fine-tuning (`T1`) on the VNWoodKnot two-class protocol. Both runs are trained for 50 epochs and then evaluated on the held-out target-domain test split. The validation trajectories over the same 50-epoch budget are reported as optimization diagnostics.

Table 2 summarizes the main runs reported in the manuscript and the role each one plays in the study. The table also makes clear that the reported runs serve different inferential roles, namely short-budget screening, extended follow-up checks, and target-domain adaptation under a fixed budget.

**Table 2:** Summary of the experiments.

| ID | Model/Change | Protocol | Epochs | Description |
|---|---|---|---|---|
| B1 | Faster R-CNN + HR MobileNet | Broader seven-class | 50 | Primary lightweight two-stage reference for the broader seven-class protocol. |
| E1 | Faster R-CNN + HR MobileNet | Curated benchmark | 50 | Primary lightweight two-stage reference for the curated benchmark. |

(Continued)

**Table 2 (continued)**

| ID | Model/Change | Protocol | Epochs | Description |
|---|---|---|---|---|
| Y0-3600 | YOLOv8s | Curated benchmark | 50 | Main short-budget curated-benchmark reference used in the focused YOLO comparison. |
| Y0-3600e200 | YOLOv8s | Curated benchmark | 200 | Final long-schedule YOLOv8s run used for literature-facing comparison on the curated benchmark. |
| Y0-full | YOLOv8s | Broader seven-class protocol | 50 | Main internal YOLOv8s reference on the broader seven-class protocol. |
| Y0-full_e200 | YOLOv8s | Broader seven-class | 200 | Long-schedule diagnostic run used to assess schedule sensitivity on the broader seven-class protocol. |
| Y1 | YOLOv8s + P2 head | Curated benchmark | 50 | Screening-stage ablation used to test whether a higher-resolution detection head improves the compact YOLO baseline. |
| Y1-e200 | YOLOv8s + P2 head | Curated benchmark | 200 | Matched long-schedule ablation used to compare the P2-head variant against the final curated-benchmark YOLO baseline. |
| Y2-e200 | YOLOv8s + WNIoU loss | Curated benchmark | 200 | Matched long-schedule ablation used to compare the WNIoU-style variant against the final curated-benchmark YOLO baseline. |
| T0 | YOLOv8s target-only training | VNWoodKnot 2-class | 50 | Target-domain baseline trained only on VNWoodKnot for supervised adaptation comparison. |
| T1 | `Y0-3600e200` → fine-tune | VNWoodKnot 2-class | 50 | Source-initialized adaptation run used to test transfer from the curated benchmark to VNWoodKnot under the same 50-epoch budget. |

### *4.2 Detector-Family Exploration*

Before focusing on YOLO-specific ablations, we compare the detector families that define the practical design space of this study.

- On the broader seven-class protocol, the high-resolution MobileNet baseline reaches 75.15% AP50, while `Y0-full` reaches 80.39% AP50.

- On the curated benchmark, the best high-resolution MobileNet variant (`E1`) reaches 74.52% AP50. Under the same benchmark family, the final compact YOLO baseline reaches 84.38% AP50 after 200 epochs.
- The low-capacity MobileNet baseline reaches 71.46% AP50 on the broader seven-class validation setting and serves mainly as an exploratory reference showing that simply keeping a lightweight Faster R-CNN + MobileNet family does not close the gap to the stronger compact YOLO baseline.

These results motivate the rest of our work: YOLOv8s becomes the main in-domain baseline not because it is the largest model we tested, but because it emerges as the strongest compact detector in our screening results across the practical wood-defect settings considered here. Table 3 shows that this detector-family gap is visible in both in-domain settings rather than being tied to only one protocol.

**Table 3:** Detector-family comparison across practical in-domain settings.

| Family/Run | Protocol | AP50 | Description |
|---|---|---|---|
| E1 | Curated benchmark | 74.52% | Exploratory low-capacity two-stage reference selected under the same in-domain protocol. |
| B1 | Broader seven-class | 75.15% | Primary lightweight two-stage reference for the broader seven-class protocol. |
| `Y0-full_e200` | Broader seven-class | 81.16% | Main compact one-stage reference for the broader seven-class protocol after the matched 200-epoch rerun. |
| `Y0-3600e200` | Curated benchmark | 84.38% | Long-schedule compact YOLOv8s run used for literature-facing comparison. |

### *4.3 YOLO Refinements and Final Curated-Benchmark Results*

Table 4 compares the broader seven-class reference, the curated-benchmark baseline, and the matched 200-epoch reruns. Figs. 2 and 3 show the class-wise AP50 profile and training dynamics of the final curated-benchmark model. The strongest result in Table 4 is `Y0-3600e200`, which reaches 84.38% AP50, 80.88% precision, and 79.11% recall on the curated benchmark. The matched broader seven-class rerun, `Y0-full_e200`, reaches 81.16% AP50, so the broader protocol remains clearly harder within the same detector family. The 50-epoch single-run references show the same protocol gap: `Y0-full` reaches 80.39% AP50, whereas `Y0-3600` reaches 83.57% AP50.

The matched 200-epoch ablations do not change that ranking. `Y1-e200` reaches 83.7% AP50 and `Y2-e200` reaches 83.93% AP50, so neither variant surpasses `Y0-3600e200`. Among these two ablations, the WNIoU-style branch comes closest to the final curated-benchmark baseline.

**Table 4:** Focused YOLO comparisons across the broader seven-class protocol, the curated benchmark, and the matched 200-epoch reruns. AP50 is the primary metric.

| ID | Dataset | Epochs | AP50 | Prec.50 | Rec.50 |
|---|---|---|---|---|---|
| `Y0-full` | Broader seven-class | 50 | 80.39% | 74.89% | 77.42% |
| `Y0-3600` | Curated benchmark | 50 | **83.57%** | 79.46% | 79.95% |
| `Y0-full_e200` | Broader seven-class | 200 | 81.16% | 76.43% | 75.85% |
| `Y1-e200` | Curated benchmark | 200 | 83.7% | 79.53% | 78.27% |
| `Y2-e200` | Curated benchmark | 200 | 83.93% | 79.23% | 79.16% |
| `Y0-3600e200` | Curated benchmark | 200 | **84.38%** | 80.88% | 79.11% |

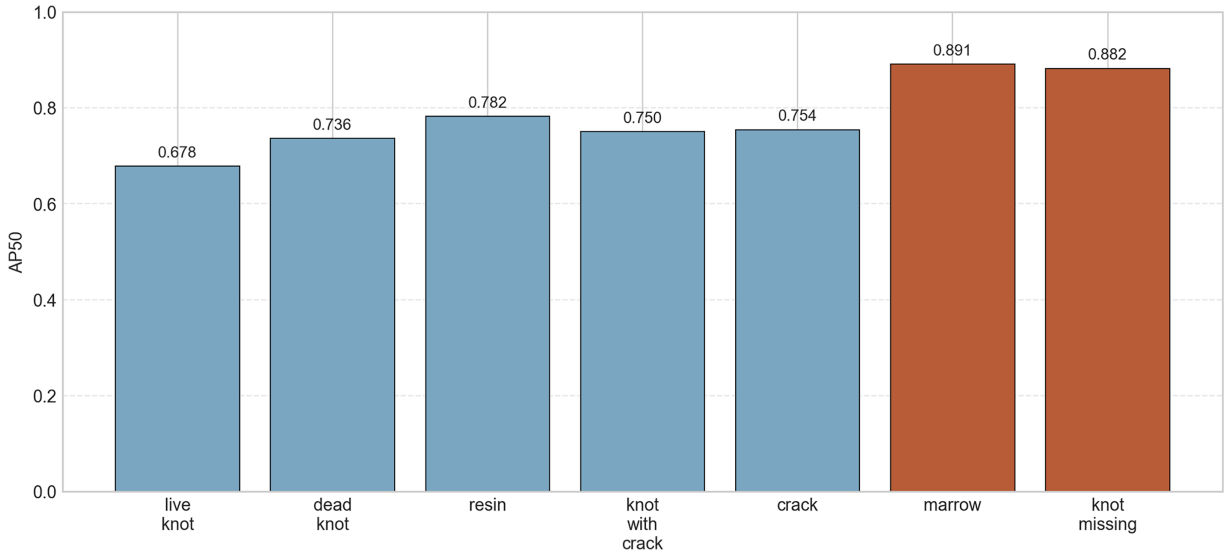


**Figure 3:** Class-wise AP50 of `Y0-3600e200`.

Among recent studies evaluated on the same benchmark family, ODCA-YOLO and SiM-YOLO reported AP50 values of 78.5% and 78.4%, respectively, while BPN-YOLO and FDD-YOLO reported 81.8% and 82.3%, respectively [2–5]. Hoang et al. reported 81.2% AP50 on a relabeled benchmark derived from the Kodytek Zenodo release using an area-attention detector [10]. Our final YOLOv8s model achieved 84.38% AP50 on the curated benchmark, placing it above the AP50 values reported in these recent studies on closely related benchmark variants.

Fig. 2 shows the training dynamics of the 200-epoch YOLOv8s run on the curated benchmark. AP50 rises quickly during the early schedule, reaches its peak at epoch 55, and then stays in a stable high-performance regime through the remainder of training. The loss curves also become progressively smoother over the schedule, which supports using this run as the final literature-facing compact baseline.

Fig. 3 shows the class-wise AP50 of `Y0-3600e200`. The strongest classes are marrow (89.1%) and knot missing (88.2%), followed by resin (78.2%). The weakest class is live knot (67.8%), while dead knot (73.6%), knot with crack (75%), and crack (75.4%) occupy the middle range. This ranking is consistent with the semantic ambiguity discussed in Section 1: knot-related categories remain harder than the most visually distinctive defect types.

### *4.4 Target-Domain Adaptation Results on VNWoodKnot*

Table 5 summarizes the target-domain adaptation comparison on VNWoodKnot. The held-out target-domain test split used here contains 229 images, 155 target boxes, and 75 negative `knot_free` images. Under the seed-42 run, `T1` reaches 83.12% AP50, whereas `T0` reaches 77.53%. Precision also rises from 53.7% to 67.3%, recall rises from 89.03% to 91.61%, and the number of predictions drops from 257 to 211. The validation trajectories show the same pattern: `T1` reaches 70% AP50 by epoch 9 and 80% by epoch 31, whereas `T0` reaches the same thresholds at epochs 19 and 40.

**Table 5:** Target-domain adaptation on VNWoodKnot under a fixed 50-epoch budget, with seed 42.

| ID | Setting | AP50 | Prec.50 | Rec.50 | Notes |
|---|---|---|---|---|---|
| `T0` | Target-only supervised training | 77.53% | 53.7% | 89.03% | Held-out test (229 images, 155 targets), 257 predictions |
| `T1` | `Y0-3600e200` → fine-tune | **83.12%** | **67.3%** | **91.61%** | Held-out test (229 images, 155 targets), 211 predictions |

This seed-42 comparison shows that the target task is learnable within a modest training budget and that source initialization can produce a stronger adapted run. The repeated-seed analysis below is used to determine whether that advantage persists across seeds. In other words, Table 5 should be read as a representative single-run comparison rather than as the final statement on transfer robustness.

Fig. 4 visualizes the same seed-42 comparison. The largest class-wise AP50 gain appears on `live_knot`, where AP50 rises from 71.06% to 83.24%. For `dead_knot`, AP50 changes little, but precision and recall still improve. The validation curves show that `T1` reaches the same accuracy range earlier than `T0`.

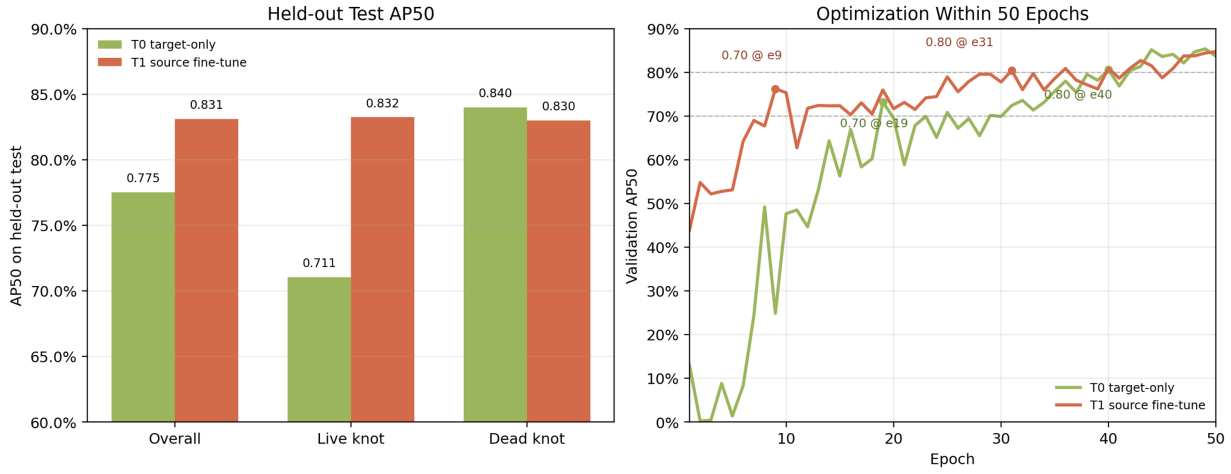


**Figure 4:** VNWoodKnot adaptation under 50-epoch budget. Left: held-out test AP50 overall and by class for target-only training and curated-benchmark source fine-tuning. Right: validation AP50 trajectories for the first 50 epochs.

Fig. 5 shows representative VNWoodKnot test examples for target-only training (`T0`) and source-initialized fine-tuning (`T1`). The selected cases illustrate both favorable and unfavorable transfer outcomes:

source initialization improves some detections, but target-only training remains competitive in other cases, and class confusion is not fully removed. This qualitative pattern matches the mixed repeated-seed target-domain results.

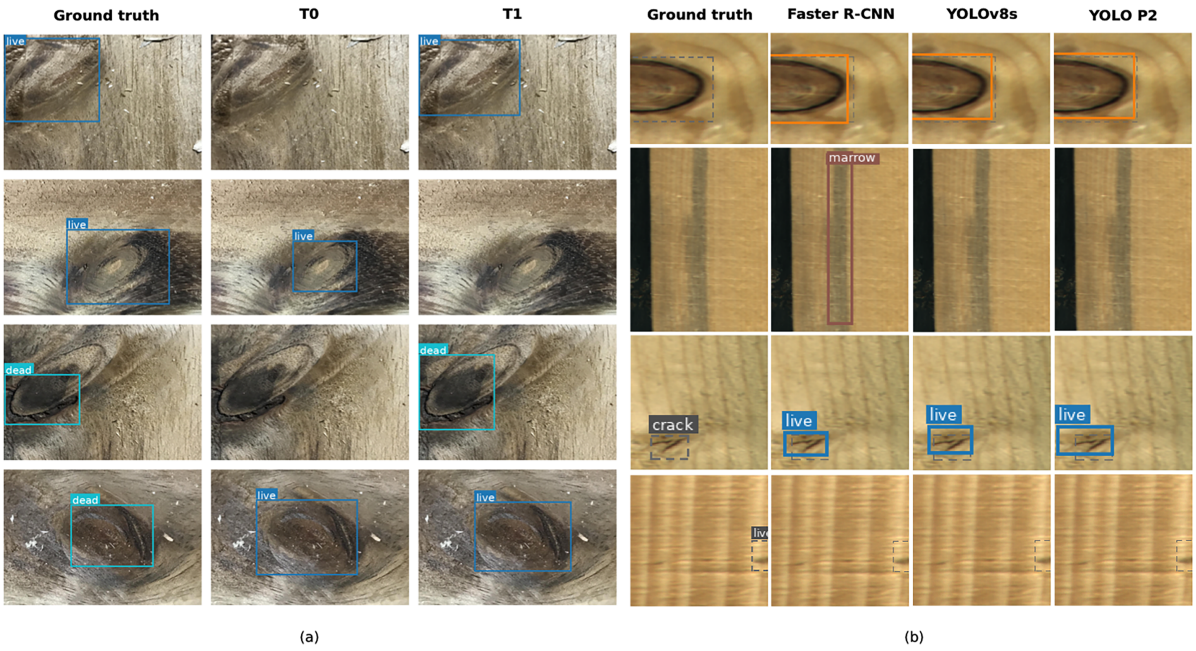


**Figure 5:** Qualitative comparisons for the target-domain and in-domain settings. The VNWoodKnot examples (**a**) show both favorable and unfavorable transfer outcomes, whereas the in-domain examples (**b**) provide supporting visual evidence for representative correct detections and failure modes across models.

### *4.5 Multi-Seed Robustness Evaluation*

Repeated-seed analyses over seeds 42, 43, 44, and 45 show that the short-budget ranking remains stable in-domain Tables A1–A3. In native YOLO validation, `Y0` reaches 83.79% ± 0.0050 AP50, `Y2` reaches 83.55% ± 0.0033, and `Y1` reaches 82.99% ± 0.0082. Under the shared repo evaluator on the held-out curated test split, the ordering becomes `Y0` (77.91% ± 0.0052) > `Y1` (77.57% ± 0.0084) > `Y2` (77.25% ± 0.0042). None of the local YOLO refinements therefore yields a robust 50-epoch held-out-test AP50 gain over the compact baseline.

The target-domain picture differs. On VNWoodKnot validation, `T1` reaches 85.35% ± 0.0101 AP50 and `T0` reaches 84.44% ± 0.0088, so source initialization remains slightly better on average under the same 50-epoch budget. On the held-out test split, however, the ordering reverses: `T0` reaches 79.09% ± 0.0136 AP50, whereas `T1` reaches 77.3% ± 0.0409. Under this budget, transfer changes optimization behavior and the precision–recall trade-off, but it does not yield a robust held-out-test AP50 gain across seeds. Taken together, Tables 6 and 7 show that the validation average and the held-out test average should not be treated as interchangeable when transfer behavior is assessed under a small target-domain budget.

### *4.6 Inference Computational Cost*

Table 8 reports inference cost for the detector families and transfer settings at image size 1024 and batch size 1. The compact YOLOv8s baseline remains attractive from a deployment perspective: despite higher FLOPs than the MobileNet baselines, it runs faster and uses less peak inference memory in the present environment.

**Table 6:** Results for target-domain adaptation under the 50-epoch setting. Reported values are mean ± standard deviation over seeds 42, 43, 44, and 45.

| Setting | AP50 | Prec.50 | Rec.50 |
|---|---|---|---|
| `T0` (target-only) | 84.4% ± 0.009 | 76.2% ± 0.037 | 82% ± 0.043 |
| `T1` (`Y0-3600e200` → fine-tune) | 85.4% ± 0.010 | 79.8% ± 0.024 | 81.9% ± 0.011 |

**Table 7:** Held-out-test repeated-seed results for target-domain adaptation on VNWoodKnot. All rows correspond to 50-epoch runs, and reported values are mean ± standard deviation over seeds 42, 43, 44, and 45.

| Setting | AP50 | Prec.50 | Rec.50 |
|---|---|---|---|
| `T0` (target-only) | 79.1% ± 0.014 | 53.9% ± 0.027 | 91.6% ± 0.017 |
| `T1` (`Y0-3600e200` → fine-tune) | 77.3% ± 0.041 | 59.9% ± 0.074 | 86.9% ± 0.032 |

**Table 8:** Inference computational cost of the detector families and transfer settings reported in the main text at image size 1024 and batch size 1.

| Model | Params (M) | FLOPs (G) | Latency (ms) | Throughput (img/s) | Peak Mem. (GB) |
|---|---|---|---|---|---|
| B1 | 18.99 | 19.64 | 9.55 | 104.70 | 0.257 |
| E1 | 18.99 | 43.33 | 12.96 | 77.14 | 0.257 |
| `Y0-3600e200` | 9.84 | 60.38 | 7.48 | 133.69 | 0.154 |
| `Y1-e200` | 10.03 | 81.24 | 9.06 | 110.39 | 0.230 |
| `Y2-e200` | 9.84 | 60.38 | 7.23 | 138.22 | 0.154 |
| `T0` | 9.84 | 60.36 | 5.98 | 167.23 | 0.154 |
| `T1` | 9.84 | 60.36 | 7.59 | 131.79 | 0.154 |

Among the YOLO variants, only the P2-head branch (`Y1-e200`) clearly increases inference cost, raising parameters, FLOPs, latency, and memory without improving AP50 over `Y0-3600e200`. The WNIoU-style branch (`Y2-e200`) keeps essentially the same inference footprint as the baseline, which is consistent with its role as a loss-level training change rather than an architectural modification. The same interpretation applies to `T0` and `T1`: source initialization changes the starting weights but not the deployed model structure, so the small latency gap between them should not be read as a structural cost difference. Table 8 therefore supports practical detector selection, but it should not be interpreted as a substitute for a full deployment or system-integration study.

## 5 Discussion

The strongest result in the evaluated design space comes from the compact YOLOv8s baseline. On the curated benchmark, the long-schedule run reaches 84.38% AP50, placing it above the AP50 values reported in recent studies on closely related benchmark variants. We do not frame this as a strict SOTA claim because the reconstructed benchmark, checkpoint policy, and evaluation details are not identical across studies.

The MobileNet-based two-stage branch provides a useful reference, especially on the broader seven-class protocol, but it does not match the compact YOLO baseline. More importantly, the local YOLO modifications do not change that conclusion. Under repeated-seed 50-epoch checks, none of the internal YOLO variants produces a robust held-out-test AP50 gain over the baseline, and the matched 200-epoch

reruns keep the same overall ranking. Within the present design space, detector-family choice therefore matters more than the tested module-level YOLO refinements.

Protocol choice also changes the observed picture enough that it cannot be treated as a reporting detail. The broader seven-class protocol remains below the curated benchmark even after the extra 200-epoch rerun, and the broader protocol does not benefit from a longer schedule in the same way as the curated benchmark. The VNWoodKnot study points in the same direction. Source initialization improves optimization and slightly improves the validation average across four seeds, yet the held-out-test aggregate favors `T0` and shows greater variance for `T1`. In other words, source-to-target transfer remains meaningful to study, but under the present 50-epoch adaptation budget it does not yield a stable held-out-test AP50 advantage across repeated seeds.

The ablation results support the same comparative conclusion. The P2-head branch remains below the compact baseline on AP50 under both short and long schedules, even though it stays competitive on mAP50–95. The WNIoU-style long-schedule rerun comes closer, but it also remains below the baseline. In this paper, these variants are therefore best interpreted as targeted diagnostic refinements within a comparative study rather than as the central claimed contribution.

This study has several limitations. The broader seven-class protocol and external adaptation results are still restricted to relatively short target-domain budgets, and the paper does not yet report matched training-time or energy measurements across schedules and detector families. The full reproducibility bundle is described here, but it is not embedded directly in the manuscript.

## 6 Conclusion

This study examined how detector-family choice, protocol breadth, and source-to-target transfer affect wood surface defect detection across the curated benchmark (VSB), a broader seven-class protocol, and the low-resource target domain.

The results allow the three research questions from the introduction to be answered directly. First, strong curated-benchmark results do not transfer unchanged when the evaluation protocol is broadened. The compact YOLOv8s baseline remains the strongest detector in the evaluated design space, but performance drops from 84.38% AP50 on the curated benchmark to 81.16% AP50 on the broader seven-class protocol, showing that protocol breadth materially changes the apparent difficulty of the task.

Second, within the present design space, detector family matters more than the tested module-level refinements. Within the 50-epoch reference setting, YOLOv8s remained ahead of the tested MobileNet-based references, and the repeated-seed short-budget checks do not overturn that ranking once held-out-test AP50 is considered. The matched 200-epoch reruns keep the same ordering on the curated benchmark: `Y1-e200` reaches 83.70% AP50 and `Y2-e200` reaches 83.93% AP50, both below the 84.38% AP50 of `Y0-3600e200`.

Third, when the model is moved to VNWoodKnot, source initialization improves optimization behavior and can strengthen a representative single run, but the repeated-seed held-out-test aggregate does not confirm a robust advantage under the present 50-epoch adaptation budget. The representative seed-42 run favored source-initialized fine-tuning, and the four-seed validation average also remained slightly higher for `T1`. The repeated-seed held-out-test aggregate, however, favored target-only training: `T0` averaged 79.09% AP50, whereas `T1` averaged 77.30% AP50.

Taken together, these findings show that benchmark breadth and source-to-target shift should be treated as central evaluation dimensions in wood surface defect detection rather than as secondary experimental details. Future work will focus on extending target-domain adaptation beyond the present 50-epoch budget,

adding matched training-time benchmarking across schedules, and studying transfer-aware adaptation strategies on VNWoodKnot.

**Acknowledgement:** Not applicable.

**Funding Statement:** The authors received no specific funding for this study.

**Author Contributions:** Khanh Nguyen-Trong conceived the study, designed the methodology, implemented the code, ran the experiments, analyzed the results, and wrote the manuscript. Tan Nguyen-Thi-Thanh generated the qualitative prediction outputs and assembled the comparative visualizations used in Fig. 5, and contributed to the writing and review of the revised manuscript. All authors reviewed and approved the final version of the manuscript.

**Availability of Data and Materials:** The source code is available in the Appendix A at Source code.

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Appendix A

This appendix reports the repeated-seed matrices underlying the robustness summaries in Section 4.5.

**Table A1:** Repeated-seed in-domain YOLO summary on the curated benchmark under the 50-epoch budget. Native val AP50 is taken from the best-checkpoint Ultralytics logs; held-out test metrics are computed with the shared evaluator. Values are mean ± standard deviation over seeds 42–45.

| Setting | Native val AP50 | Held-Out Test AP50 | Test Prec.50 | Test Rec.50 |
|---|---|---|---|---|
| `Y0` (YOLOv8s baseline) | 0.8379 ± 0.0050 | 0.7791 ± 0.0052 | 0.5491 ± 0.0118 | 0.8540 ± 0.0111 |
| `Y1` (P2 head) | 0.8299 ± 0.0082 | 0.7757 ± 0.0084 | 0.5375 ± 0.0051 | 0.8587 ± 0.0072 |
| `Y2` (WNIoU-style loss) | 0.8355 ± 0.0033 | 0.7725 ± 0.0042 | 0.5484 ± 0.0048 | 0.8523 ± 0.0047 |

**Table A2:** Per-seed held-out-test AP50 for the 50-epoch in-domain YOLO runs on the curated benchmark, evaluated with the shared evaluator. The final column reports mean ± standard deviation over seeds 42–45.

| Setting/Split | Seed 42 | Seed 43 | Seed 44 | Seed 45 | Mean ± std |
|---|---|---|---|---|---|
| `Y0` held-out test | 0.78436 | 0.77248 | 0.78163 | 0.77778 | 0.7791 ± 0.0052 |
| `Y1` held-out test | 0.78389 | 0.77715 | 0.76390 | 0.77777 | 0.7757 ± 0.0084 |
| `Y2` held-out test | 0.77147 | 0.77870 | 0.77019 | 0.76944 | 0.7725 ± 0.0042 |

**Table A3:** Per-seed held-out-test AP50 for the 50-epoch target-domain adaptation runs on VNWoodKnot, evaluated with the shared evaluator. The final column reports mean ± standard deviation over seeds 42–45.

| Setting/Split | Seed 42 | Seed 43 | Seed 44 | Seed 45 | Mean ± std |
|---|---|---|---|---|---|
| `T0` held-out test | 0.77532 | 0.78900 | 0.79072 | 0.80843 | 0.7909 ± 0.0136 |
| `T1` held-out test | 0.83119 | 0.75545 | 0.76839 | 0.73704 | 0.7730 ± 0.0409 |

## References


1. Kodytek P, Bodzas A, Bilik P. A large-scale image dataset of wood surface defects for automated vision-based quality control processes. F1000Research. 2022;10:581. doi:10.12688/f1000research.52903.1.
2. Wang R, Liang F, Wang B, Mou X. ODCA-YOLO: an omni-dynamic convolution coordinate attention-based YOLO for wood defect detection. Forests. 2023;14(9):1885. doi:10.3390/f14091885.
3. Wang R, Chen Y, Liang F, Wang B, Mou X, Zhang G. BPN-YOLO: a novel method for wood defect detection based on YOLOv7. Forests. 2024;15(7):1096. doi:10.3390/f15071096.
4. Xi H, Wang R, Liang F, Chen Y, Zhang G, Wang B. SiM-YOLO: a wood surface defect detection method based on the improved YOLOv8. Coatings. 2024;14(8):1001. doi:10.3390/coatings14081001.
5. Lema DG, Sánchez-González L, Usamentiaga R, delaCalle FJ. Benchmarking deep learning models for surface defect detection: a reproducible and statistically-rigorous approach. J Intell Manuf. 2025;126(3–4):1093. doi:10.1007/s10845-025-02672-8.
6. An H, Liang Z, Qin M, Huang Y, Xiong F, Zeng G. Wood defect detection based on the CWB-YOLOv8 algorithm. J Wood Sci. 2024;70(1):26. doi:10.1186/s10086-024-02139-z.
7. Tran V, Lam D, Le T. VNWoodKnot: a benchmark image dataset for wood knot detection and classification. Data Brief. 2025;62:112039. doi:10.1016/j.dib.2025.112039.
8. Kiliç K, Özcan U, Kiliç K, Doğru İA. Using deep learning techniques for anomaly detection of wood surface. Drv Ind. 2024;75(3):275–86. doi:10.5552/drvind.2024.0114.
9. Kılıç K, Kılıç K, Doğru İ.A, Özcan U. WD detector: deep learning-based hybrid sensor design for wood defect detection. Eur J Wood Wood Prod. 2025;83:50.
10. Hoang VT, Le VT, Dinh N, Tran-Trung K, Van BN, Thi Hong HD, et al. Deep learning-based faulty wood detection with area attention. Comput Mater Contin. 2025;85(1):1495–514. doi:10.32604/cmc.2025.066506.
11. Mittal P. A comprehensive survey of deep learning-based lightweight object detection models for edge devices. Artif Intell Rev. 2024;57(9):242. doi:10.1007/s10462-024-10877-1.
12. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst. 2015;28(6):91–9. doi:10.1109/tpami.2016.2577031.
13. Howard A, Sandler M, Chu G, Chen LC, Chen B, Tan M, et al. Searching for MobileNetV3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2019 Oct 27–Nov 2; Seoul, Republic of Korea.
14. Khan RU, Shah F, Khan AA, Tahir H. Advancing PCB quality control: harnessing YOLOv8 deep learning for real-time fault detection. Comput Mater Contin. 2024;81(1):345–67.
15. Wang Y, Huang J, Dipu MSK, Zhao H, Gao S, Zhang H, et al. YOLO-RLC: an advanced target-detection algorithm for surface defects of printed circuit boards based on YOLOv5. Comput Mater Contin. 2024;80(3):4973–95.
16. Zhao S, Li Z, Wei X, Wang Y, Zhao K. Lightweight small defect detection with YOLOv8 using cascaded multi-receptive fields and enhanced detection heads. Comput Mater Contin. 2026;86(1):1–14. doi:10.32604/cmc.2025.068138.
17. Yasir SM, Ahn H. Faster metallic surface defect detection using deep learning with channel shuffling. Comput Mater Contin. 2023;75(1):1847–61. doi:10.32604/cmc.2023.035698.
18. Wang Y, Zhu Y, Chen X, Yin T, Su S. Steel surface defect detection via the multiscale edge enhancement method. Comput Mater Contin. 2026;86(3):1–10. doi:10.32604/cmc.2025.072404.
19. Min Y, Li J, Li Y. Rail surface defect detection based on improved UPerNet and connected component analysis. Comput Mater Contin. 2023;77(1):941–62. doi:10.32604/cmc.2023.041182.
20. Wei W, Cheng Y, He J, Zhu X. A review of small object detection based on deep learning. Neural Comput Appl. 2024;36(12):6283–303. doi:10.1007/s00521-024-09422-6.
21. Hua W, Chen Q. A survey of small object detection based on deep learning in aerial images. Artif Intell Rev. 2025;58:162. doi:10.21203/rs.3.rs-3074407/v1.
22. Tong Z, Chen Y, Xu Z, Yu R. Wise-IoU: bounding box regression loss with dynamic focusing mechanism. arXiv:2301.10051. 2023. doi:10.48550/arxiv.2301.10051.

23. Wang J, Xu C, Yang W, Yu L. A normalized gaussian wasserstein distance for tiny object detection. arXiv:2110.13389. 2022. doi:10.48550/arxiv.2110.13389.
24. Zheng Z, Wang P, Liu W, Li J, Ye R, Ren D. Distance-IoU loss: faster and better learning for bounding box regression. arXiv:1911.08287. 2019. doi:10.48550/arxiv.1911.08287.