Tile-Based Detection of Apple Tree Phenological Stages with Test-Time Augmentation and Enhanced Weighted Boxes Fusion

Atsuya Sakai*[0009-0000-7673-3233], Shunsuke Shioya*[0009-0003-9613-6438], Yuta Sato*[0009-0003-2438-5654], Shoki Fukatani*[0009-0007-0278-4709], Akinori Hidaka[0000-0001-9328-2104]

Tokyo Denki University, Saitama, Japan
{26rmu11, 26rmd34, 23rd088, 26rmu15}@ms.dendai.ac.jp, hidaka.akinori@mail.dendai.ac.jp

Abstract. We describe our winning submission to the DeepPhenoTree – ECCV 2026 CVPPA Data Challenge: detecting three apple-tree phenological classes in high-resolution orchard images from four European sites. Because the targets are small and densely packed, we tile throughout: a YOLOv12l detector is trained on 1024×1024 tiles and applied to overlapping tiles under six test-time augmentation (TTA) configurations. Detections are mapped back and merged by class-wise Weighted Boxes Fusion (WBF) with a consensus term, an edge-margin filter, a nonlinear score transformation, and a class-frequency correction. The system reached 0.522 mAP@50:95 on the blind test set. An ablation shows that TTA and WBF are effective only in combination.

Keywords: Object detection, Plant phenology, Test-time augmentation, Weighted boxes fusion.

1 Introduction

The DeepPhenoTree challenge [1] requires detecting three BBCH [2] phenological stages of apple trees — flowering, fruitlet, and fruit — in orchard images from four European sites. The targets are very small and densely packed in multi-megapixel images, and adjacent stages look alike across visually different sites. We combine tile-based training and inference [3], preserving small-object features, with a consensus-oriented test-time pipeline [10] that keeps detections consistent across views.

2 Proposed Method

This section describes the proposed method, which consists of tile-based training (Sect. 2.1), the test-time pipeline (Sects. 2.2–2.5), and hyperparameter search (Sect. 2.6). The test-time components were selected empirically during the challenge based on improvements in the public leaderboard score.

2.1 Tile-Based Training

Starting from COCO-pretrained weights [11], we fine-tuned YOLOv12l [5], an attention-centric single-stage detector in the YOLO family [4]. Each image is divided into 1024×1024 tiles with 20% overlap, so that objects truncated at one tile border remain intact in a neighbor. Only tiles containing objects are exported in YOLO format; boxes retaining less than 20% of their original area are dropped to avoid training on fragments. Training used AdamW [6] (lr 2.67×10^{-4}, final lr factor 0.01, weight decay 5×10^{-4}, momentum 0.937, batch 4, input 1024) for ≤100 epochs (patience 20, 3 warm-up); it stopped at epoch 62. Augmentation followed the YOLO defaults [12] (HSV (0.015, 0.7, 0.4), hflip p=0.5, mosaic, translation 0.1, scaling 0.5). The best-validation-mAP checkpoint is used for inference.

2.2 Tile-Based Inference with Test-Time Augmentation (TTA)

Each image is processed under six TTA patterns: three isotropic scales {0.8, 0.9, 1.0} × two horizontal-flip states [17], for comparison at the fusion stage. Each rescaled image is then tiled at 1024×1024 with 30% overlap (stride 716), as in SAHI [9]; edge tiles are shifted inward. The two overlaps ended up differing for no particular reason. The detector applies only a permissive per-tile NMS (confidence 0.01, IoU 0.75, max 300). After the per-tile operations of Sect. 2.3, boxes are mapped back to original coordinates.

2.3 Edge-Margin Filter and Non-Linear Score Transform

Three operations are applied to each tile's raw detections before fusion, preventing truncated and low-confidence boxes do not bias the weighted average. (i) Low-confidence pre-filtering: detections below confidence 0.035 are discarded. (ii) Edge-margin filter (EMF): detections whose centers lie within 12% of the tile size from a tile border are removed, unless that border is the image border; such boxes are usually truncated fragments re-captured intact in a neighboring tile [14]. (iii) Non-linear score transform (ST): each score s becomes $s^{1.30}$, widening the gap between confident and borderline detections [15] and down-weighting low-confidence noise in the WBF averaging; it is inverted after fusion and the consensus boost (Sect. 2.4).

2.4 Fusion by Weighted Boxes Fusion (WBF) with a Consensus Boost

The six detection sets are fused with WBF [7] instead of suboptimal uniform averaging [13], so that each box's final score reflects how many views support it. Unlike NMS, which discards all but the highest-scoring box, WBF averages the clustered coordinates weighted by confidence, averages their scores, and rescales each fused score by $\min(N, n)/N$, where N is the number of TTA patterns producing at least one detection (≤6; patterns, not tiles) and n the raw detections in the cluster. Since n counts every raw detection rather than distinct patterns, this reweighting favors boxes backed by many overlapping proposals. As the three stages differ in size and density, fusion is per class with IoU thresholds 0.50/0.45/0.40 for flowering/fruitlet/fruit. Following consensus

voting [10], a consensus boost (CB) $s \leftarrow s\,(1 + \alpha \log(1 + n_{\text{agree}}))$ further rewards boxes with many agreeing detections; n_{agree} counts same-class raw detections across all six patterns overlapping the fused box at IoU ≥ 0.45, and $\alpha = 0.1317$.

2.5 Final Filtering and Class-Frequency Correction

After fusion and the inverse transform, detections are filtered by class-specific thresholds (0.35/0.20/0.25). Class Weighting (CW), a form of contextual rescoring [16], then multiplies each score by $w = c + (1 - c)\, r_k$, with r_k the fraction of surviving detections in the image of class k and $c = 0.4043$. This suppresses minority classes, since a single image is usually dominated by one stage. The thresholds are then reapplied, and the top 2000 boxes by score are output.

2.6 Hyperparameter Search with Optuna

The initial learning rate was set with Optuna [8] (TPE sampler; 10 trials of 5 epochs, log-uniform over $[5\times10^{-5}, 5\times10^{-4}]$, validation mAP@50 as proxy), giving 2.67×10^{-4}. Nine test-time values were likewise searched over 100 trials on validation mAP@50:95, with α, c, the overlaps, and the TTA configuration fixed. As the best values overfitted the split, we adopted a heuristically rounded configuration instead (Sects. 2.2–2.5).

3 Experiments

3.1 Dataset, Metrics, and Implementation

We used the official data [1] with the provided split (472 training, 86 validation images; the blind test size is not disclosed) and report mAP per the challenge protocol. All runs used a single YOLOv12l model, no external data or extra detectors, inside the challenge Docker image (cvppa-2026-base:v1) with fixed library versions. On one NVIDIA GeForce RTX 4090 (24 GB) with seed 0, training took 23.6 h for 62 epochs and inference 6.9 s per image.

3.2 Results

The full pipeline reached 0.543 mAP@50:95 on the validation split and 0.522 on the blind test. As the blind test data were unavailable after the challenge, we could not conduct any further analysis of test-set performance.

3.3 Ablation Study

During the challenge, the six test-time components were empirically selected and tuned, primarily by testing each component individually and retaining changes that improved

the public-leaderboard score. This procedure did not reveal interactions among components or whether the observed gains would transfer to a different evaluation split. Therefore, in this section, to systematically investigate their roles, we ablate these components over 15 unique configurations (#1–#16, where #8 and #16 are identical), all run with the 30% inference overlap of Sect. 2.2. Table 1 varies TTA, the edge-margin filter (EMF), and box merging, with ST, CB, and CW enabled; TTA off is a single view (scale 1.0, no flip), EMF off is margin ratio 0. EMF helps in all four pairs (+0.032 to +0.050), but TTA and WBF help only together, from configuration #1, TTA alone costs 0.020 (#5) and WBF alone 0.004 (#2), yet together they gain 0.015 (#6). Table 2 varies the score transform (ST; exponent 1.0 off), the consensus boost (CB; $\alpha = 0$), and class weighting (CW) with TTA, EMF, and WBF on. ST is the most effective among the three, yielding an improvement of +0.0016 on its own (+0.0015 to +0.0021). The three combined yield an improvement of +0.0020 (#9 vs #16).

Table 1. Ablation over TTA, EMF, and box merging on the validation split.

#	1	2	3	4	5	6	7	8
TTA	–	–	–	–	✓	✓	✓	✓
EMF	–	–	✓	✓	–	–	✓	✓
Box Merge	NMS	WBF	NMS	WBF	NMS	WBF	NMS	WBF
mAP@50:95	0.4965	0.4923	0.5309	0.5291	0.4761	0.5116	0.5259	0.5435

Table 2. Ablation over the score transform (ST), the consensus boost (CB), and class weighting (CW) on the validation split.

#	9	10	11	12	13	14	15	16
ST	–	–	–	–	✓	✓	✓	✓
CB	–	–	✓	✓	–	–	✓	✓
CW	–	✓	–	✓	–	✓	–	✓
mAP@50:95	0.5415	0.5419	0.5415	0.5417	0.5431	0.5434	0.5436	0.5435

4 Conclusion

We presented a tile-based YOLOv12l pipeline for apple tree phenology stage detection that combines multiple test-time components, securing first place in the challenge with 0.522 mAP@50:95. Our ablation study showed that while EMF is particularly effective on its own, TTA and WBF are highly effective in combination.

Acknowledgments

This work was partially supported by Research Institute for Science and Technology of Tokyo Denki University Grant Number Q25J-01/Japan.

References

1. Metuarea, H., Hamza, A.-D.O., Guerra, W., Zuffa, F., Panzeri, F., Patocchi, A., Lozano, L., Van Hoye, S., Laurens, F., Labrosse, J., Rasti, P., Rousseau, D.: DeepPhenoTree – Apple Edition: A Multi-site Apple Phenology RGB Annotated Dataset with Deep Learning Baseline Models. Research Square, Preprint (Version 1) (2026). https://doi.org/10.21203/rs.3.rs-8977752/v1
2. Meier, U., Graf, H., Hack, H., Hess, M., Kennel, W., Klose, R., Mappes, D., Seipp, D., Stauss, R., Streif, J., van den Boom, T.: Phenological Growth Stages of Pome Fruits (Malus domestica Borkh. and Pyrus communis L.), Stone Fruits, Currants and Strawberry. Nachrichtenblatt des Deutschen Pflanzenschutzdienstes 46, 141–153 (1994).
3. Ünel, F.O., Ozkalayci, B.O., Cigla, C.: The Power of Tiling for Small Object Detection. In: CVPRW, pp. 582–591 (2019). https://doi.org/10.1109/CVPRW.2019.00084
4. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. In: CVPR, pp. 779–788 (2016). https://doi.org/10.1109/CVPR.2016.91
5. Tian, Y., Ye, Q., Doermann, D.: YOLOv12: Attention-Centric Real-Time Object Detectors. arXiv preprint arXiv:2502.12524 (2025). https://arxiv.org/abs/2502.12524
6. Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization. In: ICLR (2019). https://arxiv.org/abs/1711.05101
7. Solovyev, R., Wang, W., Gabruseva, T.: Weighted Boxes Fusion: Ensembling Boxes from Different Object Detection Models. Image and Vision Computing 107, 104117 (2021). https://doi.org/10.1016/j.imavis.2021.104117
8. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A Next-generation Hyperparameter Optimization Framework. In: KDD, pp. 2623–2631 (2019). https://doi.org/10.1145/3292500.3330701
9. Akyon, F.C., Onur Altinuc, S., Temizel, A.: Slicing Aided Hyper Inference and Fine-tuning for Small Object Detection. In: ICIP, pp. 966–970 (2022). https://doi.org/10.1109/ICIP46576.2022.9897990
10. Casado-García, Á., Heras, J.: Ensemble Methods for Object Detection. In: ECAI. Frontiers in Artificial Intelligence and Applications, vol. 325, pp. 2688–2695 (2020). https://doi.org/10.3233/FAIA200407
11. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: ECCV, pp. 740–755 (2014). https://doi.org/10.1007/978-3-319-10602-1_48
12. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLO (Version 8.0.0). Software (2023). https://github.com/ultralytics/ultralytics
13. Shanmugam, D., Blalock, D., Balakrishnan, G., Guttag, J.: Better Aggregation in Test-Time Augmentation. In: ICCV, pp. 1194–1203 (2021). https://doi.org/10.1109/ICCV48922.2021.00125
14. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: MICCAI, pp. 234–241 (2015). https://doi.org/10.1007/978-3-319-24574-4_28
15. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.: MixMatch: A Holistic Approach to Semi-Supervised Learning. In: NeurIPS, pp. 5049–5059 (2019). https://arxiv.org/abs/1905.02249
16. Pato, L.V., Negrinho, R., Aguiar, P.M.Q.: Seeing without Looking: Contextual Rescoring of Object Detections for AP Maximization. In: CVPR, pp. 14598–14606 (2020). https://doi.org/10.1109/CVPR42600.2020.01462

17. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. In: ICLR (2015). https://arxiv.org/abs/1409.1556