DFINE-BRA: A Lightweight Algorithm for Growth Stage Detection in Brassica Crops

1st Zhimin Bai
College of Information Engineering
Northwest A&F University
Yangling, China
zhiminbai@nwafu.edu.cn

2nd Shiyi Liu
College of Information Engineering
Northwest A&F University
Yangling, China
shiyiliu@nwafu.edu.cn

3rd Huijun Yang*
College of Information Engineering
Northwest A&F University
Yangling, China
yhj740225@nwafu.edu.cn

***Abstract*—To address challenges in detecting Brassica growth stages in complex environments, this paper proposes DFINE-BRA, a lightweight detection framework. This study constructed the BRA dataset, encompassing 19 varieties and 4 growth stages, and introduced SaMam, an SSM-based augmentation method, to enhance scene diversity. The architecture integrates an MI-Block for morphological feature extraction, alongside a CME-Encoder and CSAM for multi-scale feature aggregation. To optimize efficiency, knowledge distillation and TensorRT acceleration were employed for deployment on the Jetson Orin NX platform. Experimental results show that DFINE-BRA achieves an mAP50 of 83.1% on the BRA dataset, representing a 4.6% improvement over the baseline. Notably, the model reduces parameters and GFLOPs to 40% and 32% of the original values, respectively, while achieving a real-time inference speed of 36 FPS. This study offers an efficient solution for automated Brassica monitoring on edge devices. The code is available at https://github.com/Jackbai360/DFINE-BRA.**

***Index Terms*—Brassica Growth Stages, DFINE-BRA, MI-Block, CME-Encoder, Knowledge Distillation**

I. INTRODUCTION

Brassica is a globally significant genus within the Brassicaceae family, encompassing diverse cultivars used as vegetables, oilseeds, and forage. Human-driven selection since the Roman era has produced immense interspecies diversity [1]–[3], making accurate growth stage identification essential for precision management and optimal harvesting. However, genetic breeding still relies on subjective, inefficient manual labeling, which is prone to data loss. This study leverages deep learning to develop an efficient, real-time system for precise Brassica growth stage monitoring.

To accurately identify crop growth stages and improve maturity detection precision, traditional research has largely relied on hand-crafted features and machine learning. For instance, studies have employed PCA for color-based classification of oil palm [4], a two-stage SVM-KNN strategy for blueberry maturity [5], and discriminant analysis for apricot fruit ripeness [6]. While these methods advanced detection technology, they rely on manually designed features that struggle to capture deep representations, resulting in insufficient generalization in complex natural environments.

To address these challenges, researchers have focused on constructing scene specific datasets to improve recognition. Recent works developed specialized datasets for grapes covering multiple growth stages [7], strawberries across diverse climates [8], and nocturnal blueberry harvesting via illumination enhancement [9]. Despite these efforts, such methods typically incur high collection and annotation costs and lack the flexibility to adapt to highly volatile climatic conditions.

Deep learning has further pushed the boundaries of growth stage detection. Modern approaches include the lightweight YOLO-fungi model [10], a hybrid YOLOv12 based neural network for tomato detection on smartphones [11], and an efficient scale-aware RT-DETR model for pomegranate recognition [12]. However, these models often face a trade-off: lightweight architectures are frequently not validated on resource-constrained edge devices, while high-performance Transformer-based models suffer from substantial computational overhead that hinders continuous, real-time monitoring.

Modern research struggles to balance high performance with lightweight design, while existing methods often lack adaptability to specific scene variations due to data constraints. Consequently, enhancing model robustness against environmental changes and balancing architectural efficiency remain critical challenges for automated Brassica monitoring. To address this, we propose a detection model based on D-FINE [13] for real-time, accurate growth stage tracking. Our main contributions are:

- To address the issue of insufficient scene coverage, the SaMam image augmentation method is introduced to simulate feature-level environmental interference, complementing the newly constructed BRA dataset which encompasses 76 categories across 19 Brassica varieties and 4 growth stages.
- The DFINE-BRA algorithm is proposed to enhance morphological feature extraction and capture complex growth variations. By integrating the MI-Block, CME-Encoder, and CSAM, the model significantly strengthens multi-scale feature representation and cross-level interaction.

*Corresponding author: Huijun Yang (yhj740225@nwafu.edu.cn)
This work was supported by the National Natural Science Foundation of China (No. 32472002); The "Four-Chain Integration Program" of the QinChuangyuan Industrial Innovation Cluster in Shaanxi Province (No. 2025CY-JJQ-166); and the Science and Technology Program Project of Yangling Demonstration Zone (No. 2025CYFZ-07).

- To balance lightweight deployment and detection accuracy, a logits-based knowledge distillation strategy is adopted, substantially reducing parameters and computational overhead. Combined with TensorRT optimization, real-time monitoring is achieved on the Jetson Orin NX platform.

II. Materials and Dataset

A. Data Acquisition

Experimental samples for this study were collected from the Experimental Demonstration Farm of the Shenzhen Agricultural Technology Promotion Center and various online platforms. Based on phylogenetic relationships [3] and following variety identification by on-site experts, a total of 16 varieties of Chinese Cabbage (*Brassica pekinensis*), 2 varieties of Chinese Kale (*Brassica oleracea*), and 1 variety of Leaf Mustard (*Brassica juncea*) were selected as data sources. Each variety covers four critical growth stages: seedling, growing, bolting, and flowering; specific variety names are detailed in Fig. 1. To enhance generalization, the dataset comprises 1920×1080 RGB and video-extracted images of Brassica samples captured from three perspectives (front, $45°$ oblique, and top-down) across diverse indoor and outdoor environments.

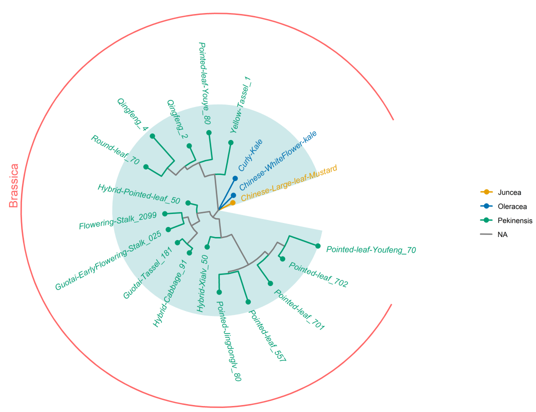

Fig. 1. Phylogenetic tree of brassica species.

The Brassica varieties and growth stages were annotated using the X-AnyLabeling [14] tool. Upon completion of the annotation, label files were exported in COCO JSON format, totaling 3,816 initial images. The dataset was subsequently partitioned into training, validation, and test sets according to an 8:1:1 ratio. Following this, SaMam was applied to perform a 4-fold style transfer augmentation for each set.

B. Data Augmentation

To enhance the model's generalization and robustness in complex natural environments, this study introduces the Style-aware Mamba (SaMam) [15], a style-aware encoder-decoder architecture, to perform style transfer-driven data augmentation on the Brassica growth stage dataset; the underlying principle is illustrated in Fig. 2. Specifically, the Mamba encoder first extracts content features from the Brassica images, while a series of representative weather style images (e.g., dust storm, cloudy, rainy, foggy) are selected as style inputs to extract style embeddings. During the decoding phase, the Style-aware Visual State Space Module (SAVSSM) dynamically generates style-related state-space parameters to achieve adaptive modulation of content features by style information. Built upon the Mamba State Space Model (SSM) [16], SaMam leverages the advantages of linear computational complexity and a global effective receptive field. This allows it to efficiently transfer diverse stylistic features to target images while preserving structural integrity, effectively simulating scene variations of Brassica species under different weather conditions without requiring re-annotation, thereby significantly reducing the workload.

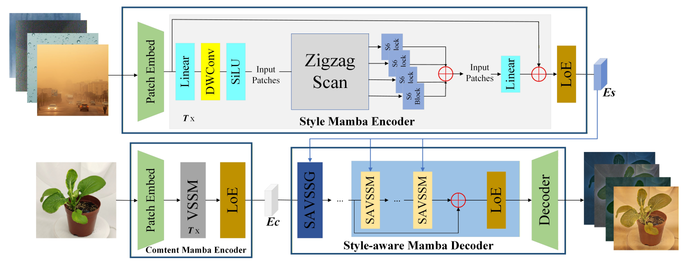

Fig. 2. Schematic illustration of SaMam image augmentation [15].

III. The Proposed DFINE-BRA Method

A. DFINE-BRA

While D-FINE was originally designed for general object detection, this study introduces targeted improvements to the DFINE-S version to meet the specific requirements of Brassica growth stage detection, resulting in the DFINE-BRA, whose structure is illustrated in Fig. 3. The backbone replaces the original HG-Blocks with MI-Blocks, which leverage multi-branch structures and depthwise separable convolutions to enable the efficient parallel extraction and semantic fusion of multi-scale features, such as leaf textures and inflorescence morphology. Additionally, to counteract the loss of sensitivity toward critical morphological features in deeper layers, an innovative Cascaded Multi-scale Enhancement Encoder (CME-Encoder) replaces the original encoder. Utilizing progressive feature diffusion strategies, the CME-Encoder effectively fuses multi-scale features from low to high levels, preserving fine edge details while strengthening the modeling of global plant architecture and spatial context.

B. MI-Block

The backbone network of the original D-FINE, HGNetv2 [17], relies on its core component, the HG-Block, which utilizes static convolutional kernels and lacks the capacity to perceive temporal morphological transitions. The MI-Block integrates a multi-gradient flow synergistic enhancement mechanism with the efficient multi-scale [18] and structural optimization concepts of InceptionNeXt [19], constructing a feature extraction unit that excels in both semantic hierarchy representation and computational efficiency. The structure is

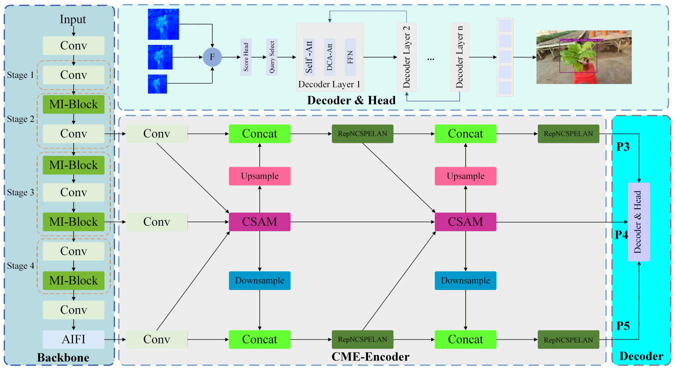

Fig. 3. Overall architecture of DFINE-BRA.

shown in Fig. 4. Specifically, it employs convolutions for cross-channel feature recalibration and dimensionality reduction, combined with depthwise separable convolutions to maintain spatial feature extraction capabilities while minimizing parameter count. Furthermore, multi-level cascading and shortcut connections are utilized to facilitate gradient flow and feature reuse. This design not only introduces rich feature interactions along the path from input to output but also enhances gradient diversity through structural differences between branches, thereby achieving more thorough and stable optimization during training. As shown in Fig. 5, the improved MI-Block achieves clearer extraction of Brassica leaf edges and veins, while also strengthening the differentiation between the background and crop contours during the bolting and flowering stages.

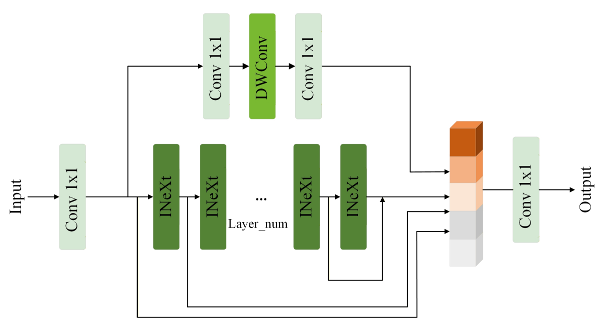

Fig. 4. Structure of the MI-Block.

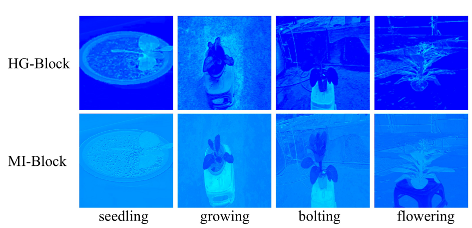

Fig. 5. Featuremap visualization (Demonstrating differences in feature extraction capabilities.)

C. CME-Encoder

To enhance multi-scale representation and cross-level interaction for Brassica growth detection, this study designs the Cascaded Multi-scale Enhancement Encoder (CME-Encoder). Traditional FPN and PANet-based encoders rely on linear paths with limited interaction depth, struggling to achieve sufficient nonlinear fusion between shallow details and deep semantics. This bottleneck hinders the representation of objects with varying sizes and ambiguous morphologies, such as stamens.

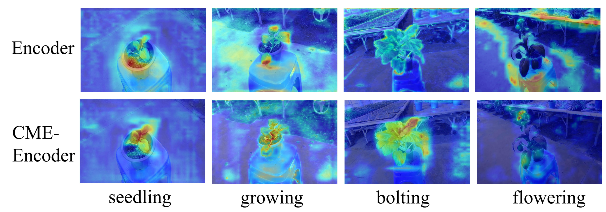

Fig. 6. Heatmap visualization (Representing differences in model attention focus areas.)

To overcome this bottleneck, the CME-Encoder introduces a Cross-Scale Aggregation Module (CSAM). Through an iterative bidirectional feature refinement mechanism, it achieves deeper and more efficient integration of multi-scale features. As illustrated in Fig. 6, this approach effectively enhances the model's ability to focus on key morphological features, such as leaves and flowers. Furthermore, it further strengthens the global spatial modeling capability of the plant, enabling the model to ignore various interferences within the complex scene.

D. CSAM

In the CME-Encoder architecture, the Cross-Scale Aggregation Module (CSAM) serves as the core innovative unit. It employs an Inception-style design to enhance input features across multiple receptive fields, as shown in Fig. 7, specifically tailored to process features from three distinct scales: $P3$,$P4$, and $P5$.

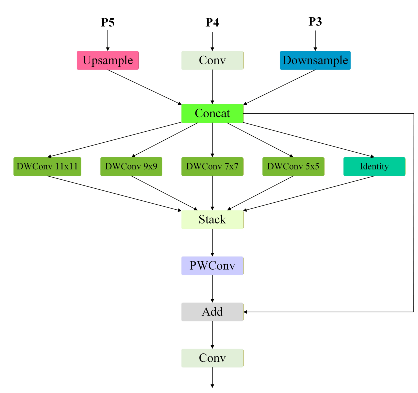

Fig. 7. Structure of the CSAM.

CSAM begins by partitioning the input feature maps. Since features at different scales exhibit significant disparities in spatial dimensions, an alignment process is first performed. The $P3$ layer features, which possess the strongest semantic information but lack fine details, are downsampled via convolution to achieve spatial alignment. The $P4$ layer features are processed directly through a convolutional layer. Conversely, the $P5$ layer features, which contain the richest spatial detail but have a higher resolution, are first upsampled to twice their size and then passed through a 1×1 convolution to adjust the channel dimensions, matching the target fusion scale. Subsequently, these processed features are concatenated and further refined by a set of depthwise separable convolutions with varying kernel sizes to extract multi-scale characteristics. Its core computational form can be expressed as (1):

$$F_{\text{out}} = F_{\text{in}} + \left(\sum_{s \in S} \mathcal{W}s \right) * F_{\text{in}} + \sum_{s \in S} \delta_s. \tag{1}$$

Here, $\mathcal{W}_s$ represents convolutional kernels of different sizes($S = 5, 7, 9, 11$), δ_s is the corresponding offset. F_{in} denotes the input feature maps, F_{out} represents the concatenated feature mapping and the resulting fused output feature. By utilizing parallel convolutional operations with varying receptive fields, this structure simultaneously captures local details and global contextual information, ensuring that target features are fully extracted across different scales. The application of depthwise separable convolutions also substantially reduces computational overhead.

IV. Experimental Results and Analysis

A. Experimental Setup

The experimental setup is shown in Table I.

TABLE I
Experimental Environment Configuration

Category	**Configuration**	**Category**	**Configuration**
System	Ubuntu 22.04	PyTorch	2.8.0
CPU	AMD EPYC 7K62	Python	3.10
GPU	NVIDIA RTX 4090	CUDA	12.8
VRAM	48 GB	Image size	640 × 640
RAM	512 GB	Optimizer	AdamW
Batch Size	8	Learning Rate	0.0002
Epochs	160	Weight Decay	0.0001

B. Evaluation Indicators

This study employs GFLOPs (Giga Floating Point Operations), Params (number of model parameters), and mAP (mean Average Precision) at Intersection over Union (IoU) thresholds of 0.5, 0.75, and 0.5–0.95 as evaluation metrics. Lower GFLOPs and Params values indicate lower computational overhead, making the model suitable for resource-constrained edge deployment. Conversely, higher mAP values indicate superior detection performance at the corresponding IoU thresholds. (2)-(5) present the specific formulas.

$$Precision = \frac{TP}{TP + FP} \times 100\%, \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \times 100\%, \tag{3}$$

$$AP = \int_0^1 P(R)\, dR, \tag{4}$$

$$mAP = \frac{1}{C} \sum_{i=1}^{C} AP_i. \tag{5}$$

C. Ablation Experiment

To validate the proposed modules' effectiveness, ablation studies were conducted based on the DFINE-S model, using HGNetv2 as the backbone network. Different module configurations were compared by adding or removing specific components stepwise. The experimental results are summarized in Table II.

TABLE II
Ablation Study Results on the BRA Test Set

Model	**HGNetv2**	**MI-Block**	**CSAM**	**Params**	**GFLOPs**	**mAP50**
DFINE-S	✓			**10.2M**	**25.2**	0.785
+MI-Block	✓	✓		13.1M	30.0	0.790
+CSAM	✓		✓	12.0M	32.1	0.816
DFINE-BRA	✓	✓	✓	14.9M	37.3	**0.823**

As shown in Table II, the MI-Block enables more precise capture of subtle Brassica variety features, enhancing the backbone ability to distinguish between different varieties and growth stages in complex scenarios, resulting in a 1.5% increase in mAP50. By synchronously fusing adjacent and intra-layer features, the CSAM achieves dynamic integration of cross-scale information, effectively replenishing spatial details missing in shallow features. Ultimately, the overall model achieves an mAP50 of 82.3% with an increase of around 46% in parameter count.

D. Comparative Experiments

To comprehensively validate the effectiveness and generalization capability of the proposed DFINE-BRA algorithm, this study conducted a series of comparative experiments on the BRA and COCO2017 [20] datasets. The performance of DFINE-BRA was evaluated against current mainstream algorithms, including Conditional-DETR [21], Deformable-DETR [22], RT-DETR [17], RT-DETRv2 [23], and the YOLOv11 [24], YOLOv12 [25], and YOLOv13 [26] series.

As shown in Table III and Table IV, compared with other models of the same DETR architecture, DFINE-BRA achieves the highest detection accuracy while maintaining the minimum number of parameters and computational cost. Although DFINE-BRA has a slightly higher computational overhead compared to the YOLO series models, the performance improvements it provides are significant enough to compensate for this drawback. Under the more stringent metric, DFINE-BRA reaches 68.1% on the self-built dataset and 49.1% on the

TABLE III
COMPARATIVE EXPERIMENTS ON THE BRA TEST SET

Model	Params	GFLOPs	mAP50	mAP75	mAP50-95
Conditional-DETR	44.0M	195.0	0.794	0.703	0.630
Deformable-DETR	40.0M	173.0	0.781	0.710	0.612
RT-DETR-R18	20.0M	60.0	0.720	0.672	0.584
RT-DETRv2-S	20.0M	60.0	0.808	0.729	0.650
YOLO11-S	9.4M	21.5	0.783	0.696	0.625
YOLOv12-S	9.3M	21.4	0.764	0.699	0.621
YOLOv13-S	**9.0M**	**20.8**	0.805	0.732	0.645
D-FINE-S	10.2M	25.2	0.785	0.719	0.646
DFINE-BRA(Ours)	14.9M	37.3	**0.823**	**0.762**	**0.681**

TABLE IV
COMPARATIVE EXPERIMENTS ON THE COCO2017 VALIDATION SET

Model	Params	GFLOPs	$\mathrm{AP}^{\mathrm{val}}_{50}$	$\mathrm{AP}^{\mathrm{val}}_{75}$	$\mathrm{AP}^{\mathrm{val}}_{50\text{-}95}$
Conditional-DETR	44.0M	195.0	0.654	0.485	0.451
Deformable-DETR	40.0M	173.0	0.652	0.500	0.462
RT-DETR-R18	20.0M	60.0	0.638	0.504	0.465
RT-DETRv2-S	20.0M	60.0	0.651	0.521	0.481
YOLO11-S	9.4M	21.5	0.634	0.503	0.466
YOLOv12-S	9.3M	21.4	0.642	0.510	0.471
YOLOv13-S	**9.0M**	**20.8**	0.652	0.520	0.480
D-FINE-S	10.2M	25.2	0.656	0.526	0.485
DFINE-BRA(Ours)	14.9M	37.3	**0.661**	**0.533**	**0.491**

COCO2017 dataset, outperforming all comparison models and demonstrating superior detection precision and robustness.

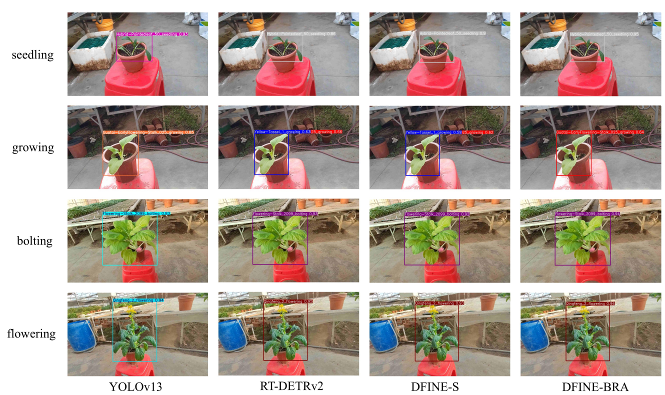

Fig. 8. Comparison of recognition capabilities among typical object detection algorithms.

As illustrated in Fig. 8, compared to the original D-FINE and RT-DETRv2 models, DFINE-BRA demonstrates stronger feature extraction and recognition capabilities across the growth stages of Brassica crops, effectively reducing bounding box overlap. While it slightly trails CNN-based YOLO models in certain scenarios, DFINE-BRA outperforms YOLO during the bolting and flowering stages, where plant phenotypic features are more abundant, achieving higher prediction accuracy. This highlights the effectiveness of the proposed improvements.

V. LIGHTWEIGHT DESIGN AND DEPLOYMENT

A. Knowledge Distillation

Knowledge distillation (KD) enhances student model performance by using teacher outputs as supervisory signals [27]. In object detection, this process primarily involves classification and bounding box distillation, employing binary classification and IoU-based localization losses, respectively. The distillation loss is calculated using (6) to (11).

$$\mathcal{L}_{\text{BCE}}\left(p^s_{i,j}, p^t_{i,j}\right) = -\left[\left(1-p^t_{i,j}\right)\log\left(1-p^s_{i,j}\right) + p^t_{i,j}\log\left(p^s_{i,j}\right)\right], \tag{6}$$

$$\mathcal{L}^{\text{dis}}_{\text{cls}}(\mathbf{x}) = \sum_{i=1}^{n}\sum_{j=1}^{K}\mathcal{L}_{\text{BCE}}\left(p^s_{i,j}, p^t_{i,j}\right), \tag{7}$$

$$w_{i,j} = \left|p^t_{i,j} - p^s_{i,j}\right|, \tag{8}$$

$$\mathcal{L}^{\text{dis}}_{\text{cls}}(\mathbf{x}) = \sum_{i=1}^{n}\sum_{j=1}^{K} w_{i,j}\mathcal{L}_{\text{BCE}}\left(p^s_{i,j}, p^t_{i,j}\right), \tag{9}$$

$$\mathcal{L}^{\text{dis}}_{\text{loc}}(\mathbf{x}) = \sum_{i=1}^{n}\max_{j}\left(w_{i,j}\right)\left(1-u'_i\right), \tag{10}$$

$$\mathcal{L}_{\text{BCKD}}(\mathbf{x}) = \alpha_1\mathcal{L}^{\text{dis}}_{\text{cls}}(\mathbf{x}) + \alpha_2\mathcal{L}^{\text{dis}}_{\text{loc}}(\mathbf{x}). \tag{11}$$

The knowledge distillation process is governed by the BCKD loss, which selectively transfers expertise from the teacher to the student by focusing on prediction gaps. Specifically, the classification distillation loss $\mathcal{L}^{\text{dis}}_{\text{cls}}(\mathbf{x})$ is defined as a sum of Binary Cross-Entropy (BCE) terms weighted by $w_{i,j}$, an approach that forces the student to prioritize learning from samples where its predictions significantly deviate from the teacher's soft targets. To further refine spatial precision, the localization distillation loss $\mathcal{L}^{\text{dis}}_{\text{loc}}(\mathbf{x})$ incorporates the maximum discrepancy $\max_j(w_{i,j})$ to scale the localization error $(1-u'_i)$, ensuring that regions with high classification uncertainty receive stronger structural supervision. The final objective is a multi-task formulation, $\mathcal{L}_{\text{BCKD}}(\mathbf{x})$, which provides a balanced optimization signal for both categorical accuracy and bounding box/pixel-level alignment.

B. Embedded Deployment

To validate the deployment of DFINE-BRA, this study utilized the NVIDIA Jetson Orin NX (16GB) under Ubuntu 22.04 using NVIDIA TensorRT. The trained PyTorch model was converted to ONNX and optimized via C++ TensorRT to build an efficient inference engine. Benchmarking was conducted in FP32 precision with a batch size of 1.

This study evaluates the performance of the model after lightweight optimization in terms of detection accuracy and inference speed based on the BRA dataset. The experimental results are shown in Fig. 9. Compared to the original teacher model DFINE-BRA, the distilled student model DFINE-BRA(KD), with a lighter network structure, achieves a 0.8% increase in mAP50 while reducing the parameter count by 72% and GFLOPs by 78%. On the Jetson Orin NX, the accelerated KD+TensorRT model achieves a single-frame inference time

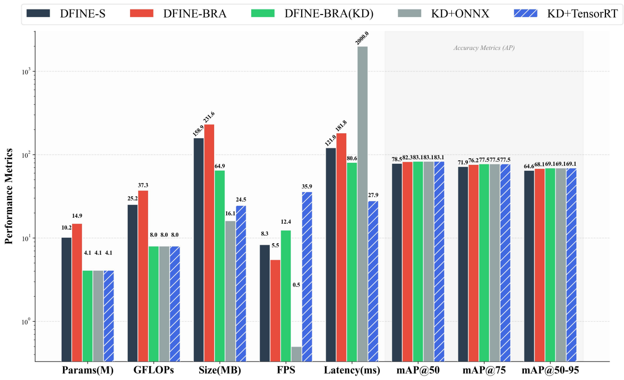

Fig. 9. Performance comparison after knowledge distillation and TensorRT acceleration (The vertical axis presented in logarithmic scale.)

of 27.9 ms, an 52.7 ms reduction compared to the non-accelerated DFINE-BRA(KD) model, resulting in a frame rate of 35.9 FPS. This fulfills the requirements for real-time detection without sacrificing accuracy. The final model size is only 15% of the original DFINE-S model, demonstrating the superior deployment advantages of the proposed model on embedded platforms.

VI. Conclusion

To address the demand for automated growth stage detection in Brassica crops, this study presents a lightweight deployment solution. By integrating SaMam for scene enrichment, the MI-Block for morphological feature extraction, and the CME-Encoder for enhanced cross-level feature fusion, the proposed framework achieves superior detection performance. Optimized via knowledge distillation and TensorRT, the model became significantly lighter and faster. Achieving real-time speeds, the algorithm demonstrates significant potential for deployment on edge devices like agricultural UAVs. Future work will focus on optimizing detection capabilities for dense scenarios and small targets in large-scale cultivation environments, such as breeding factories and open fields, to further expand the application scenarios of the algorithm.

References

[1] T. Wang, A. D. van Dijk, X. Cai, J. Wu, G. Bonnema, and X. Wang, "Brassica diversity through the lens of polyploidy: genomic evolution, introgression, and homoeologous exchange," *Horticultural Plant Journal*, 2025.

[2] P. Subramanian, H. Brijesh, and B.-S. Hahn, "Phylogenetic patterns of understudied species within the plant genus brassica: Future prospects in the post genomics-era," *Plant Gene*, vol. 41, p. 100489, 2025.

[3] J. Wang, Y. He, Y. Zhang *et al.*, "A study on origin, evolution and spread of brassica in china," *Zhongguo Nongxue Tongbao.*, vol. 22, no. 8, pp. 489–494, 2006.

[4] A. Septiarini, A. Sunyoto, H. Hamdani, A. A. Kasim, F. Utaminingrum, and H. R. Hatta, "Machine vision for the maturity classification of oil palm fresh fruit bunches based on color and texture features," *Scientia Horticulturae*, vol. 286, p. 110245, 2021.

[5] K. Tan, W. S. Lee, H. Gan, and S. Wang, "Recognising blueberry fruit of different maturity using histogram oriented gradients and colour features in outdoor scenes," *Biosystems engineering*, vol. 176, pp. 59–72, 2018.

[6] M. Khojastehnazhand, V. Mohammadi, and S. Minaei, "Maturity detection and volume estimation of apricot using image processing technique," *Scientia Horticulturae*, vol. 251, pp. 247–251, 2019.

[7] Y. Li, C. Mo, G. Wang, J. Huang, and W. Song, "Grape detection at different growth stages in complex orchard environments using state space models," *Computers and Electronics in Agriculture*, vol. 237, p. 110743, 2025.

[8] D. Xu, Z. Li, C. Zhang, G. Ouyang, Y. Chen, W. Xu, R. Qu, F. Lin, and S. Tan, "Real-time detection of multi-growth-stage strawberry in complex orchard environments via smartphone," *Smart Agricultural Technology*, p. 101468, 2025.

[9] R. Zhang, W. Dong, P. Hou, H. Li, X. Han, Q. Chen, F. Li, and X. Zhang, "Yolov11-bsd: Blueberry maturity detection under simulated nighttime conditions evaluated with causal analysis," *Smart Agricultural Technology*, p. 101314, 2025.

[10] W. Zhang, X. Wang, J. Huang, and H. Yang, "Yolo-fungi: Detection method for the growth stages of edible fungi under complex environments," in *2025 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2025, pp. 1–9.

[11] K. Khabarlak, I. Laktionov, I. Udovyk, and G. Diachenko, "Smartphone tomato growth monitoring using yolo12 neural network," in *2025 15th International Conference on Advanced Computer Information Technologies (ACIT)*. IEEE, 2025, pp. 832–835.

[12] S. Wang, S. Dong, S. Chen, and M. Liu, "An efficient scale-aware model based on the improved rt-detr for pomegranate growth stage detection," *Neurocomputing*, vol. 647, p. 130462, 2025.

[13] Y. Peng, H. Li, P. Wu, Y. Zhang, X. Sun, and F. Wu, "D-fine: Redefine regression task in detrs as fine-grained distribution refinement," *arXiv preprint arXiv:2410.13842*, 2024.

[14] W. Wang, "Advanced auto labeling solution with added features," https://github.com/CVHub520/X-AnyLabeling, CVHub, 2023.

[15] H. Liu, L. Wang, Y. Zhang, Z. Yu, and Y. Guo, "Samam: Style-aware state space model for arbitrary image style transfer," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 28 468–28 478.

[16] T. Dao and A. Gu, "Transformers are ssms: Generalized models and efficient algorithms through structured state space duality," *arXiv preprint arXiv:2405.21060*, 2024.

[17] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "Detrs beat yolos on real-time object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 16 965–16 974.

[18] Y. Feng, J. Huang, S. Du, S. Ying, J.-H. Yong, Y. Li, G. Ding, R. Ji, and Y. Gao, "Hyper-yolo: When visual object detection meets hypergraph computation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 47, no. 4, pp. 2388–2401, 2024.

[19] W. Yu, P. Zhou, S. Yan, and X. Wang, "Inceptionnext: When inception meets convnext," in *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, 2024, pp. 5672–5683.

[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[21] D. Meng, X. Chen, Z. Fan, G. Zeng, H. Li, Y. Yuan, L. Sun, and J. Wang, "Conditional detr for fast training convergence," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 3651–3660.

[22] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[23] W. Lv, Y. Zhao, Q. Chang, K. Huang, G. Wang, and Y. Liu, "Rt-detrv2: Improved baseline with bag-of-freebies for real-time detection transformer," *arXiv preprint arXiv:2407.17140*, 2024.

[24] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *arXiv preprint arXiv:2410.17725*, 2024.

[25] Y. Tian, Q. Ye, and D. Doermann, "Yolov12: Attention-centric real-time object detectors," *arXiv preprint arXiv:2502.12524*, 2025.

[26] M. Lei, S. Li, Y. Wu, H. Hu, Y. Zhou, X. Zheng, G. Ding, S. Du, Z. Wu, and Y. Gao, "Yolov13: Real-time object detection with hypergraph-enhanced adaptive visual perception," *arXiv preprint arXiv:2506.17733*, 2025.

[27] L. Yang, X. Zhou, X. Li, L. Qiao, Z. Li, Z. Yang, G. Wang, and X. Li, "Bridging cross-task protocol inconsistency for distillation in dense object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 17 175–17 184.