Supplementary Material for LED-Net: A Lightweight and Efficient Dual-Branch Convolutional Neural Network for High-Performance Fruit Tree Branches Semantic Segmentation on Mobile Devices

1. Cross-Domain Generalization Evaluation of LED-Net for Apple Branch Segmentation

1.1. Cross-Year and Intra-Regional Generalization Evaluation

To comprehensively evaluate the generalization capability of LED-Net under varying temporal and spatial conditions, we performed a cross-year and intra-regional generalization test. An independent external dataset was collected from multiple sub-orchards in Liquan County, Xianyang City, Shaanxi Province during September–October 2023. In contrast to the training dataset acquired in 2024, this new dataset differs significantly in collection year and spatial distribution of tree populations. Additionally, it introduces greater variations in lighting, tree morphology, and background complexity, thereby simulating more realistic and challenging orchard environments.

To ensure the reliability and consistency of evaluation, all images were manually annotated following the same labeling protocol as the training data, thus eliminating annotation-induced bias in performance comparisons.

From this dataset, 273 images were randomly selected to form a validation subset, designated as **Apple Branch Seg-Val2023**. This dataset is publicly available at https://github.com/ly27253/LED-Net. The LED-Net model, trained solely on the original dataset, was directly evaluated on this validation set without any fine-tuning. This strategy enables an objective assessment of the model's robustness in previously unseen but intra-regional domains.

Representative visual results are illustrated in Fig. 1. Despite pronounced differences in orchard layout, illumination, and branch morphology, LED-Net accurately segments tree branches with clear structural boundaries, demonstrating strong

adaptability to unseen intra-regional conditions. Quantitative comparisons with existing lightweight segmentation models, including PIDNet, DDRNet, BiSeNetV1/V2, STDC1/2, SegNeXt, and HRNet, are provided in Table 1. LED-Net achieves the highest Intersection over Union (IoU: 73.52%) and F1-score (84.74%) among all methods, while also maintaining superior recall (81.80%). These results confirm LED-Net's capability for accurate and stable segmentation across intra-regional domains without requiring domain-specific adaptation.

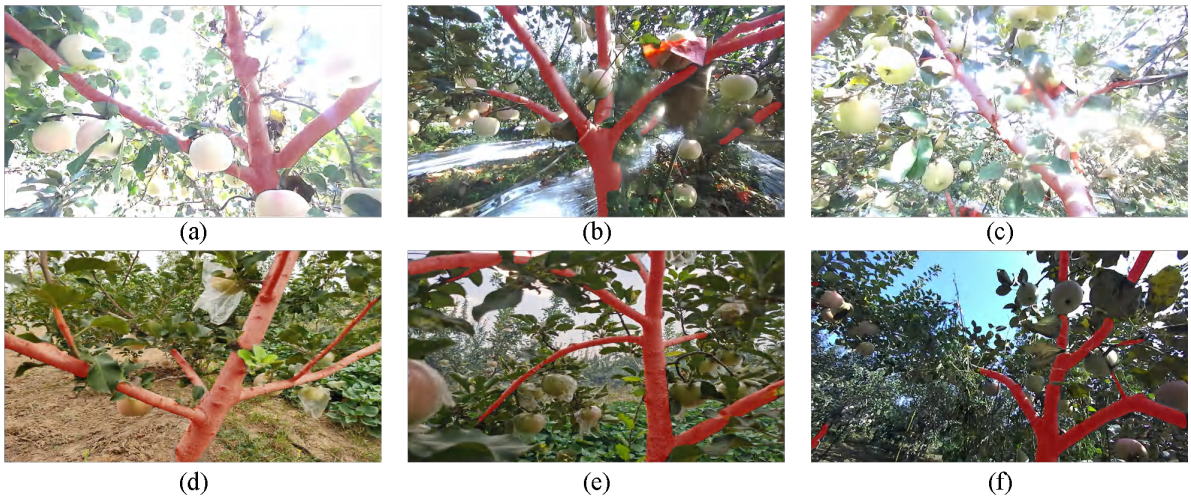

Fig. 1. Semantic segmentation results of LED-Net on Apple Branch Seg-Val2023 (intra-region, cross-year test).

1.2. Cross-Regional and Cross-Seasonal Generalization Evaluation

To further assess the model's resilience under substantial spatiotemporal domain shifts, we constructed an external test set collected during May–June 2021 in orchards located in Jingbian County, Yulin City, Shaanxi Province. This dataset exhibits strong domain discrepancy relative to the training set (2024) in terms of geographical region, climate, tree species, and phenological growth stage. With a spatial separation of over 500 kilometers and a temporal gap exceeding one year, this test scenario presents a highly challenging generalization task involving both cross-regional and cross-seasonal variation.

Manual annotation was performed consistently with prior protocols, using Photoshop to ensure high-quality semantic labels. A total of 324 images were randomly selected to constitute the **Apple Branch Seg-Val2021** evaluation set, which is publicly

Table 1: Segmentation performance comparison on Apple Branch Seg-Val2023 dataset (intra-region, cross-year generalization). **Bold** indicates best, underline is second-best.

Model	IoU (%)	Acc (%)	F1-score (%)	Precision (%)	Recall (%)
BiSeNetV1	68.03	73.51	80.97	**90.11**	73.51
HRNet	63.50	68.90	77.67	89.01	68.90
STDC1	64.24	69.59	78.23	89.31	69.59
STDC2	65.54	71.52	79.19	88.69	71.52
BiSeNetV2	72.90	79.66	84.33	89.58	79.66
SegNeXt	59.09	63.53	74.28	89.41	63.53
DDRNet	71.01	77.62	83.08	89.38	77.62
PIDNet	73.28	80.22	84.58	89.44	80.22
Ours (LED-Net)	**73.52**	**81.80**	**84.74**	87.90	**81.80**

accessible at https://github.com/ly27253/LED-Net.

The LED-Net model trained exclusively on the original dataset was tested directly on this set without fine-tuning. Table 2 summarizes the comparative segmentation metrics against other mainstream lightweight networks. Expectedly, due to increased background complexity and pronounced morphological differences, all models showed decreased performance, particularly in structural measures such as IoU and F1-score.

Despite the increased difficulty, LED-Net consistently surpasses competing approaches, demonstrating enhanced IoU (54.77%) and F1-score (70.78%). These results evidence the model's strong structural representation ability and segmentation resilience in the presence of significant spatiotemporal variations, reinforcing its applicability for practical agricultural deployment. Representative qualitative segmentation outcomes are presented in Fig. 2.

Overall, these evaluations confirm that LED-Net achieves superior transferability and robust structural perception across diverse unseen domains, highlighting its potential for broad agricultural scene generalization.

Table 2: Comparison of segmentation performance on Apple Branch Seg-Val2021 dataset (cross-region, cross-season evaluation). **Bold** indicates best, underline is second-best.

Model	IoU (%)	Acc (%)	F1-score (%)	Precision (%)	Recall (%)
BiSeNetV1	52.65	57.54	68.98	86.10	57.54
HRNet	40.56	43.12	57.71	87.25	43.12
STDC1	33.25	34.69	49.90	**88.88**	34.69
STDC2	44.97	49.25	62.04	83.79	49.25
BiSeNetV2	48.90	52.29	65.68	88.30	52.29
SegNeXt	45.95	48.92	62.97	88.32	48.92
DDRNet	52.57	58.50	68.91	83.83	58.50
PIDNet	48.23	54.34	65.08	81.10	54.34
Ours (LED-Net)	**54.77**	**61.41**	**70.78**	83.52	**61.41**

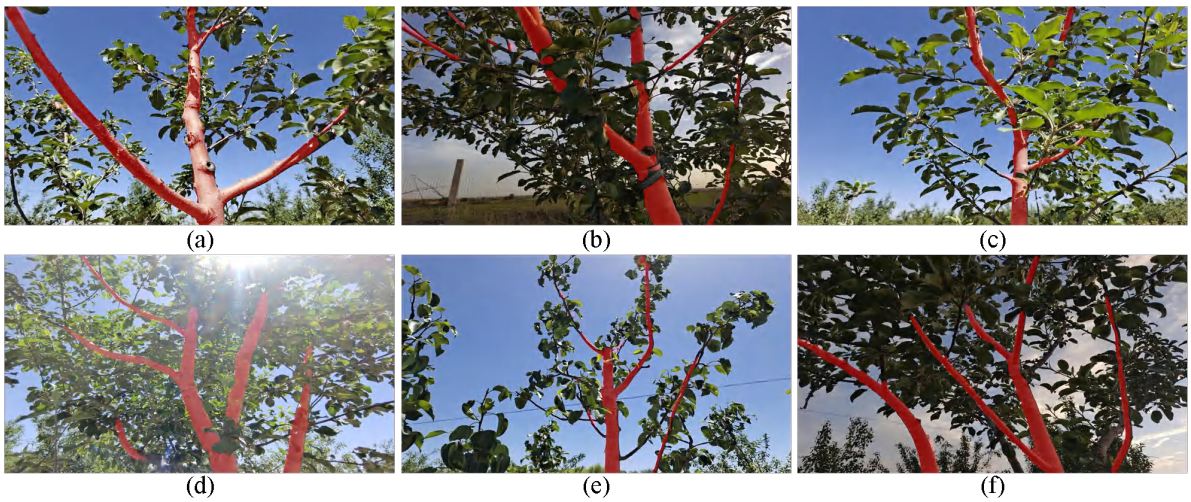

Fig. 2. Semantic segmentation results of LED-Net on Apple Branch Seg-Val2021 (cross-region, cross-season evaluation).

2. Analysis of Functional Integration Between the Transformer Module and CNN Branch in the GETB

We investigate the functional integration mechanism between the Transformer module and the CNN branch within our GETB architecture. Contrary to a conventional dual-branch design, the component identified as the CNN branch consists primarily of skip connections derived from the Transformer input features. These skip features are element-wise added to the Transformer output prior to further convolutional refinement, indicating that the fusion occurs between early input features and the Transformer's output rather than between two parallel and independent branches.

To evaluate the effectiveness of this fusion strategy, Grad-CAM visualizations were performed on feature maps from three critical stages, as shown in Fig. 3. The first column displays the input features passed through the skip connection, characterized by rich spatial details but scattered activation patterns. The second column presents the Transformer module output, which highlights semantically coherent and globally contextualized regions. The third column illustrates the fused feature maps obtained via element-wise summation, effectively combining spatial localization with semantic context.

The visualization results reveal that the input features maintain fine-grained spatial information but suffer from fragmented activations and limited semantic coherence. In contrast, the Transformer output produces more focused and semantically meaningful responses over target regions. The fused results demonstrate enhanced activation coverage across target structures while suppressing irrelevant noise, confirming the complementary nature of these two feature sources.

To quantitatively assess the contribution of this fusion mechanism, an ablation study was conducted by removing the skip connections, relying solely on the Transformer output. The outcomes, reported in Table 3, show a noticeable degradation in key metrics such as Intersection over Union (IoU) and F1-score, confirming the fusion's positive impact on model performance. Furthermore, an additional ablation experiment, removing both the skip connections and the subsequent convolutional layer in the GETB tail, further reduced performance slightly, supporting the impor-

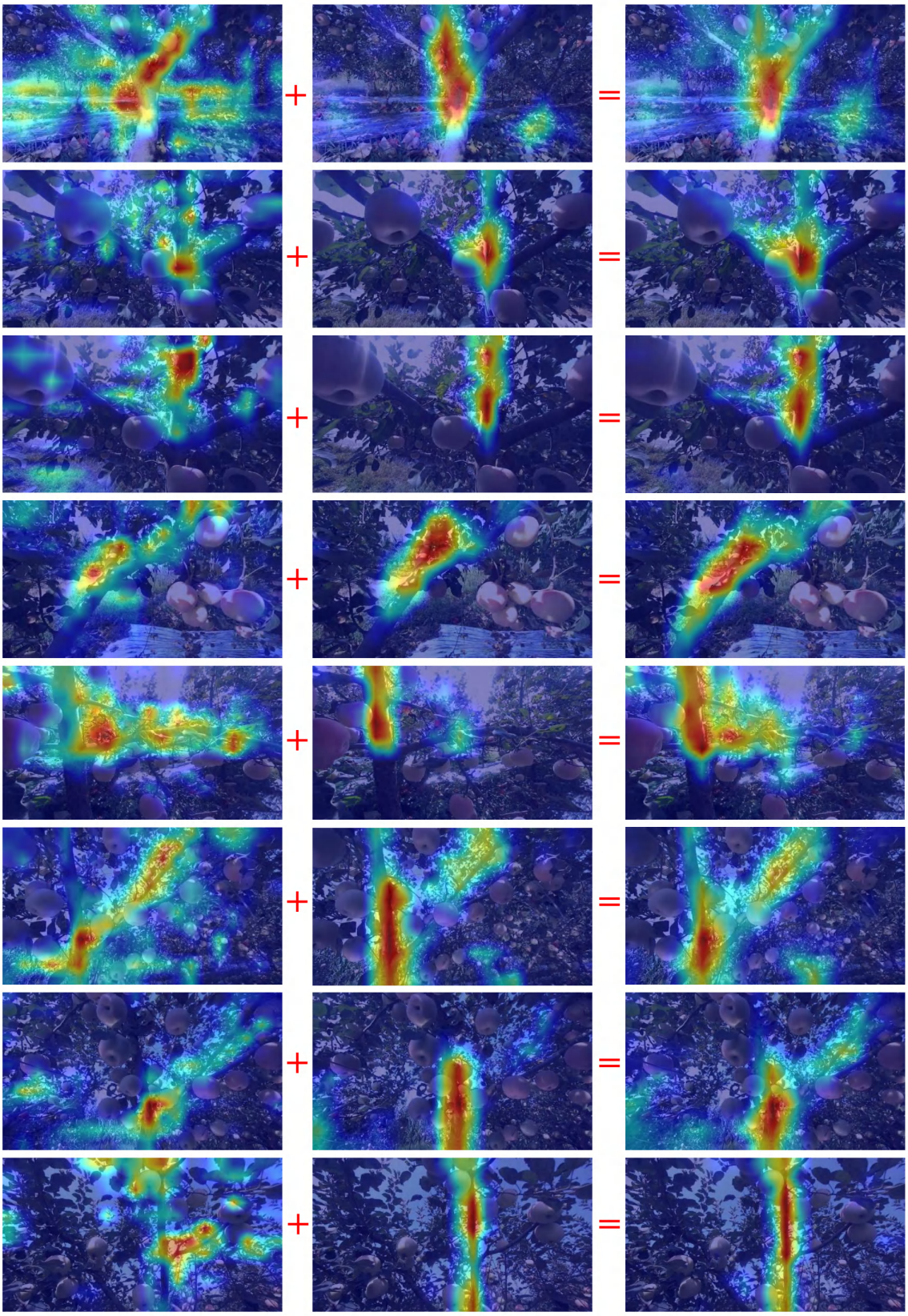

Fig. 3. Grad-CAM visualizations of three fusion stages across multiple samples. Columns from left to right show: (1) skip connection input—rich in spatial detail but with dispersed activations; (2) Transformer output—highlighting semantically coherent regions; and (3) fused features—combining spatial and semantic cues via element-wise summation.

tance of convolutional refinement for semantic consistency.

Table 3: Ablation study on the fusion strategy and convolutional refinement within the GETB module.

Model	IoU (%)	Acc (%)	F1-score (%)	Precision (%)	Recall (%)
LED-Net	**81.46**	90.13	**89.78**	**89.43**	90.13
w/o Skip Connection (in GETB)	81.09	**90.15**	89.43	88.73	**90.15**
w/o Skip Connection & Conv (in GETB)	81.03	90.03	89.52	89.01	90.03

Additionally, visualization of the convolutional layer following the Transformer module was performed via Grad-CAM to explore its effect on feature representation. Fig. 4 shows that the attention regions before and after convolution remain largely consistent, with the post-convolution heatmaps exhibiting a slight intensification of activations. This observation suggests that the convolutional refinement subtly enhances and consolidates the attention map produced by the Transformer without altering its semantic focus.

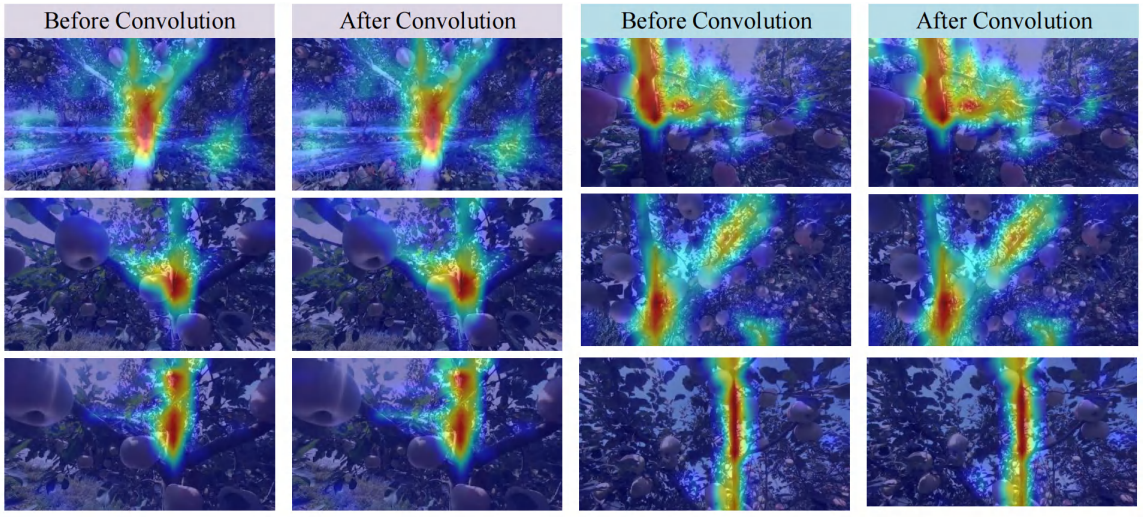

Fig. 4. Grad-CAM visualizations before and after the convolutional layer in the GETB, with only subtle changes observed.

In summary, both qualitative and quantitative analyses demonstrate that the element-wise summation fusion strategy effectively integrates local fine-grained features from the skip connections with the Transformer's global semantic context. This integration

enhances the overall representation quality and mitigates potential semantic inconsistencies introduced by post-Transformer convolutional layers. These findings provide strong empirical evidence supporting the design choices in the GETB module.

3. Analysis of LED-Net Training Dynamics and Model Capacity

In this supplementary material, we present a detailed analysis of the training dynamics and capacity of LED-Net, motivated by concerns regarding its limited accuracy improvement despite significantly reduced parameters and FLOPs. This investigation aims to provide a deeper understanding of the model's optimization behavior and generalization ability.

As is commonly observed in deep learning, there exists an intrinsic trade-off between model accuracy and computational complexity. Achieving high performance with extremely low computational cost is inherently challenging. However, through further optimization informed by constructive insights, we improved LED-Net's Intersection over Union (IoU) from the previously reported **81.05%** to **81.46%**. This improvement notably widens the performance margin compared to **BiSeNetV1 (80.61%)** and **DDRNet (80.62%)** benchmarks, demonstrating the effectiveness of our refined model design.

To investigate potential underfitting caused by the model's compact architecture, we examined the training loss curves for *Spatial Loss*, *Context Loss*, and their combined loss function, defined as $\text{Loss}_{\text{total}} = \text{Spatial Loss} + 0.4 \times \text{Context Loss}$, consistent with the revised manuscript formulation. As illustrated in Fig. 5, all three losses show a rapid decrease during the early training phase, followed by gradual convergence to stable values, indicating effective learning and the absence of underfitting phenomena.

Furthermore, we tracked key validation metrics—IoU and F1-score—throughout the training process, as shown in Fig. 6. Both metrics exhibit rapid increases within the initial 3000 iterations, followed by steady improvements until approximately 57000 iterations, where the curves plateau, indicating model convergence and robust generalization capacity.

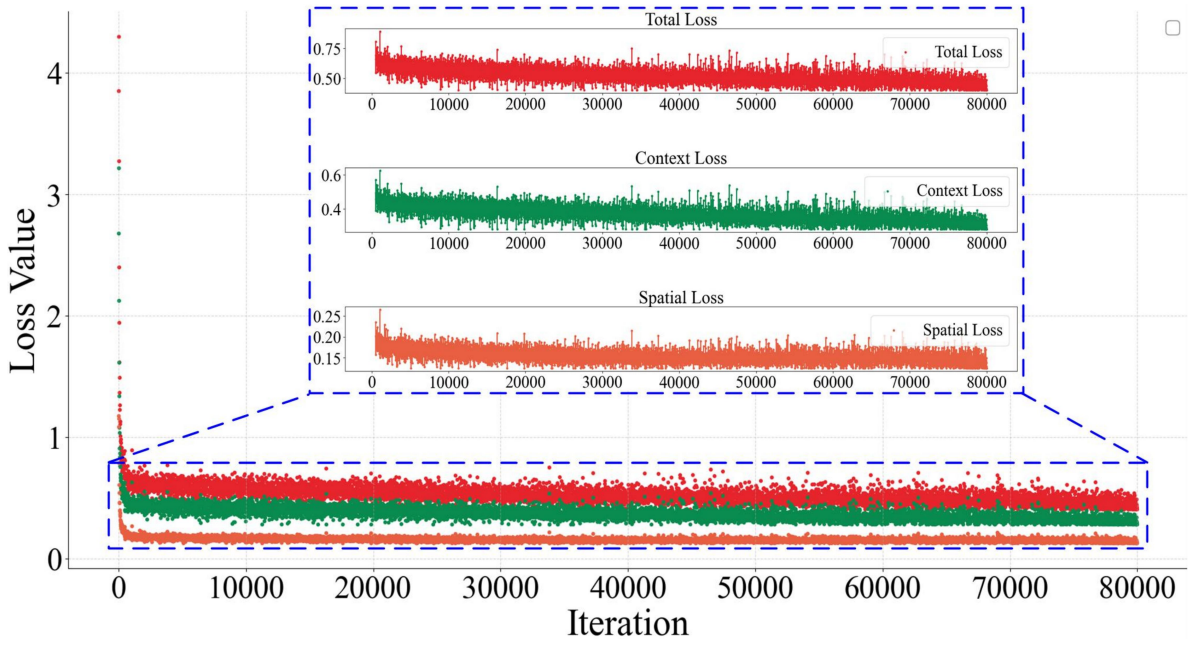

Fig. 5. Training loss curves for LED-Net, including *Spatial Loss*, *Context Loss*, and their combined loss ($\text{Loss}_{\text{total}} = \text{Spatial Loss} + 0.4 \times \text{Context Loss}$).

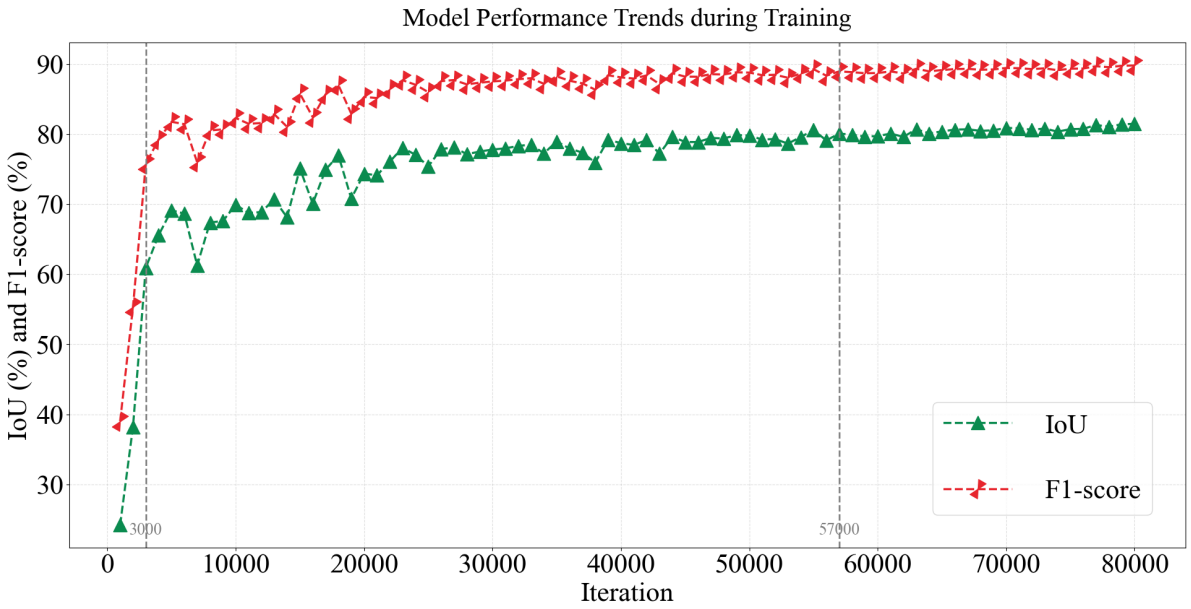

Fig. 6. Validation IoU and F1-score curves during the training process.

Overall, these analyses confirm that LED-Net's compact architecture maintains sufficient capacity for effective training without underfitting or overfitting, supporting its competitive accuracy relative to larger models while significantly reducing computational demands.

The comprehensive training dynamics and performance evaluation presented herein complement the main manuscript and provide valuable insights for future model design and optimization strategies. Due to manuscript space constraints, the detailed quantitative data and code will be released publicly alongside our implementation.

4. Analysis of Threshold Selection for Binarizing Edge Features in SEAM

The threshold used for binarizing edge features in SEAM plays a critical role in determining the segmentation performance. However, this parameter was not clearly specified or justified in the main manuscript. To address this gap, we conducted a comprehensive study on the impact of threshold selection, exploring both fixed-value thresholds and adaptive percentile-based thresholding strategies.

4.1. Fixed Threshold Evaluation

We first evaluated the model performance with fixed thresholds varying from 0.1 to 0.9 in increments of 0.1. Table 4 summarizes the results in terms of Intersection over Union (IoU), Accuracy (Acc), F1-score, Precision, and Recall. The highest IoU and F1-score were achieved with a threshold of 0.6, indicating this value as an effective fixed threshold for edge binarization.

4.2. Percentile-Based Adaptive Thresholding

Recognizing that edge response distributions differ across images, a fixed threshold may be suboptimal in some cases. To address this, we proposed an adaptive thresholding mechanism based on percentiles of edge responses within each image. The threshold T is defined as the smallest value satisfying

$$T = \inf \left\{ t \in \mathbb{R} \,\middle|\, \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{1}_{\{B_{\text{initial}}(i,j) \leqslant t\}} \geqslant p \right\} \tag{1}$$

where $B_{\text{initial}}(i,j)$ denotes the edge response at pixel (i,j), H and W represent

Table 4: Performance comparison using different fixed thresholds (0.1 to 0.9) for binarizing edge features. **Bold** indicates best.

Fixed Threshold	IoU (%)	Acc (%)	F1-score (%)	Precision (%)	Recall (%)
0.1	81.05	90.57	89.53	88.52	90.57
0.2	80.53	89.63	89.22	88.81	89.63
0.3	81.04	90.00	89.53	89.06	90.00
0.4	80.85	89.79	89.41	89.09	89.79
0.5	80.78	89.82	89.37	88.93	89.82
0.6	**81.16**	89.63	**89.60**	**89.57**	89.63
0.7	80.96	**90.92**	89.48	88.08	**90.92**
0.8	80.95	90.22	89.47	88.74	90.22
0.9	81.05	89.61	89.53	89.45	89.61

height and width of the feature map, and $p \in [0, 1]$ corresponds to the desired percentile threshold. This approach dynamically adjusts T to accommodate varying edge response distributions.

Table 5 presents the performance evaluation over percentiles ranging from 10% to 90%. The 80th percentile threshold demonstrated the best performance, achieving an IoU of 81.46% and an F1-score of 89.78%.

Table 5: Performance comparison using different percentile thresholds (10% to 90%) for binarizing edge features. **Bold** indicates best.

Percentile Threshold	IoU (%)	Acc (%)	F1-score (%)	Precision (%)	Recall (%)
10%	81.06	90.49	89.54	88.61	90.49
20%	81.03	90.82	89.52	88.26	90.82
30%	80.97	89.76	89.49	89.21	89.76
40%	81.34	90.27	89.71	89.16	90.27
50%	81.26	90.07	89.66	89.25	90.07
60%	81.16	89.89	89.60	89.31	89.89
70%	81.09	90.80	89.56	88.35	90.80
80%	**81.46**	90.13	**89.78**	**89.43**	90.13
90%	80.96	**91.04**	89.48	87.97	**91.04**

4.3. Comparison of Thresholding Strategies

Figure 7 visualizes the segmentation performance trends comparing fixed-value and percentile-based thresholds. The percentile-based method exhibits superior robustness and adaptability, effectively addressing the variability in edge responses across different images.

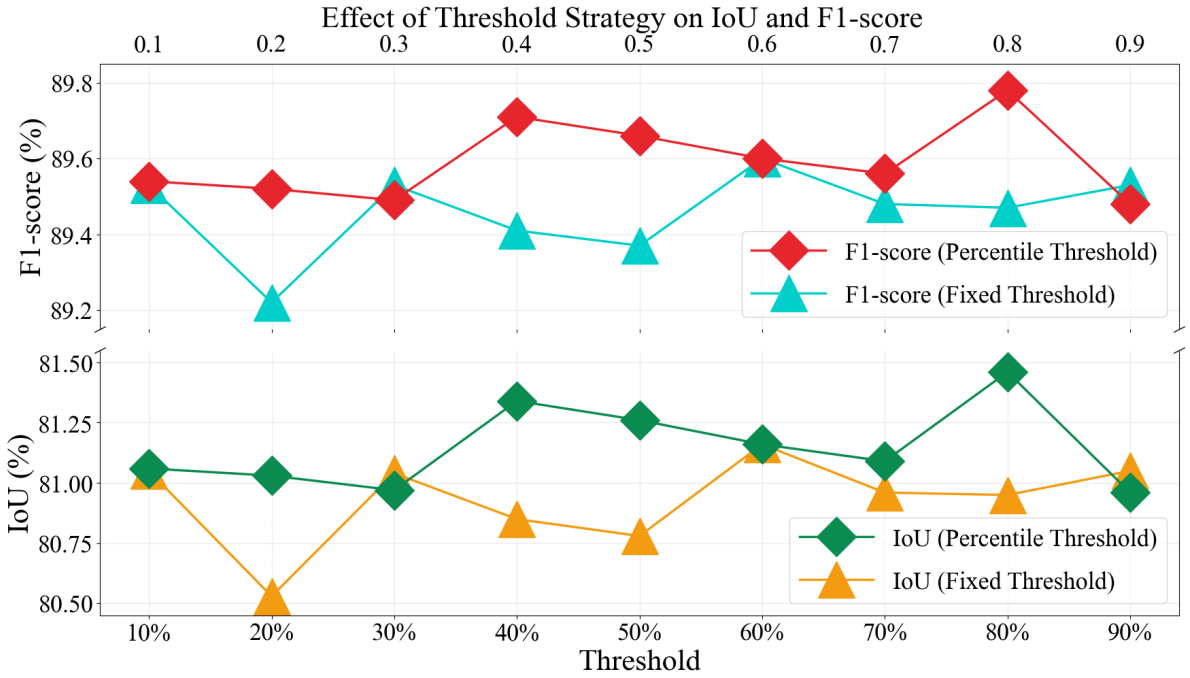

Fig. 7. Comparison of segmentation performance under fixed-value thresholds versus percentile-based thresholds.

This analysis confirms that while a fixed threshold of 0.6 performs well generally, adopting a dynamic percentile-based threshold significantly improves segmentation robustness and accuracy. We therefore recommend integrating the percentile-based thresholding strategy into the SEAM framework to enhance the reliability of edge feature binarization.

All relevant results and discussions have been incorporated into the revised manuscript. The presented supplementary study provides a rigorous foundation for the choice of thresholding method in edge-based segmentation tasks.

5. Real-Time Inference Performance Evaluation on Low-Power Edge Platforms

To comprehensively assess the real-time inference capabilities of our model under practical deployment conditions, we conducted additional experiments focusing on its performance on representative low-power GPU platforms commonly used in mobile robotics. These supplementary evaluations address deployment scenarios that were not fully covered in the original manuscript, specifically the model's behavior on edge computing hardware.

5.1. Supplementary Experiment 1

The original PyTorch model was tested on an MSI CREATOR P50 laptop equipped with an Intel i5-12400F CPU and an NVIDIA GTX1660S GPU. This setup serves as an onboard computing unit for our laboratory's mobile robot and operates independently, powered by a 19V mobile power supply from the robot platform. With an input resolution of 1280×720, the model achieved an inference speed of approximately 57.5 FPS, which closely matches the ZED camera's maximum frame rate of 60 FPS at the same resolution. This confirms that the model meets real-time requirements for typical mobile robot applications. Detailed results for all tested models and configurations are summarized in Table 6.

5.2. Supplementary Experiment 2

To improve inference speed further, the model was exported to ONNX format and subjected to a series of optimizations including static graph conversion and deployment using ONNX Runtime and TensorRT. These optimizations yielded substantial acceleration without loss of segmentation accuracy. On a high-end RTX3090 GPU, the optimized ONNX model demonstrated significant speed improvements, illustrating the potential for efficient deployment after optimization. It is important to note that due to the architecture and implementation specifics, the SegNext model could not be successfully converted to ONNX format. Comparative FPS data are included in Table 6.

5.3. Supplementary Experiment 3

The optimized ONNX model was then deployed on the low-power GTX1660S GPU platform, resulting in an increased inference speed of approximately 75.39 FPS. This performance gain further confirms the model's suitability and efficiency for edge deployment in resource-constrained environments (see Table 6).

Table 6: Unified comparison of inference speed (FPS) under different deployment strategies (original PyTorch implementation vs. optimized ONNX version) and hardware platforms (RTX3090 and GTX1660S).

Model	FPS			
	PyTorch, RTX3090	PyTorch, GTX1660S	ONNX, RTX3090	ONNX, GTX1660S
BiSeNetV1	112.64	32.68	144.76	63.62
HRNet	65.14	27.63	78.35	45.18
STDC1	147.19	57.51	190.74	82.26
STDC2	117.03	51.66	138.57	61.49
BiSeNetV2	102.71	37.65	173.46	62.31
SegNeXt	42.13	33.89	–	–
DDRNet	203.62	87.96	261.32	105.51
PIDNet	162.10	75.87	197.25	82.35
Ours (LED-Net)	126.02	57.50	177.49	75.39

For clearer visualization and comparison, the original PyTorch-based inference speeds on the RTX3090 and the supplementary experimental results are consolidated in Fig. 8.

Summary: The additional experiments confirm that, after optimization, our model maintains excellent real-time inference performance even on low-power GPU platforms typical of edge computing scenarios. This verifies the model's practical feasibility for deployment in mobile and resource-constrained robotic systems.

6. Supplementary Study on Dilation Rate Configurations in the SESP Module

In the design of the SESP module, the choice of dilation rates in the spatial and semantic branches plays a critical role in balancing spatial detail preservation and contextual information aggregation. To systematically investigate the impact of different

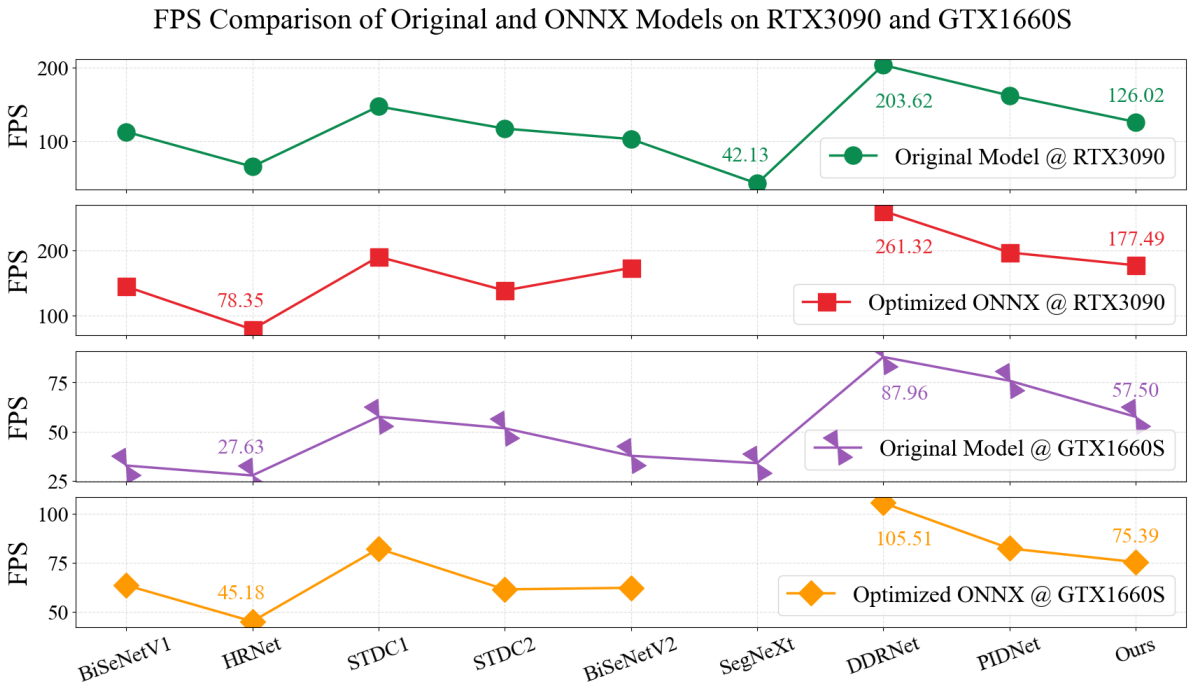

Fig. 8. Unified comparison of inference speeds across different segmentation models and deployment scenarios. The figure consolidates results from the original manuscript (PyTorch implementation on RTX 3090) and the three supplementary experiments: (1) ONNX-optimized models on RTX 3090, (2) PyTorch implementation on GTX1660S, and (3) ONNX-optimized models on GTX1660S.

dilation rate configurations, we conducted an extensive comparative study involving a variety of dilation settings.

Specifically, we evaluated a series of representative dilation rate patterns in both branches, including uniform values, progressively increasing sequences, decreasing sequences, and irregular combinations. These configurations were chosen to capture diverse receptive field behaviors and to assess their influence on segmentation performance metrics.

The results indicate that the configuration with a fixed dilation rate of [1,1,1,1] in the spatial branch combined with progressively increasing dilation rates [1,2,3,4] in the context branch achieves the highest overall performance, notably in IoU and F1-score. This pattern facilitates effective multi-scale context aggregation without sacrificing local spatial feature integrity. The progressive dilation rates in the context branch expand the receptive field gradually, enabling the network to capture hierarchical semantic information while mitigating gridding artifacts. Simultaneously,

Table 7: Comparison of model performance under different dilation rate configurations in the spatial and semantic branches of the SESP module. **Bold** indicates best.

Dilation Rates		IoU (%)	Acc (%)	F1-score (%)	Precision (%)	Recall (%)
Spatial Branch	Context Branch					
[1,1,1,1]	[1,2,3,4]	**81.46**	90.13	**89.78**	89.43	90.13
[1,1,1,1]	[1,1,1,1]	81.10	**90.78**	89.57	88.38	**90.78**
[1,1,1,1]	[1,3,5,7]	81.20	90.06	89.62	89.19	90.06
[1,1,1,1]	[2,2,2,2]	81.23	89.62	89.64	**89.67**	89.62
[1,1,1,1]	[2,3,6,9]	80.93	90.45	89.46	88.49	90.45
[1,2,3,4]	[1,2,3,4]	81.39	90.02	89.74	89.47	90.02
[1,2,3,4]	[1,1,1,1]	81.33	90.01	89.70	89.40	90.01
[1,3,5,7]	[1,1,1,1]	81.18	89.94	89.61	89.28	89.94
[2,2,2,2]	[1,1,1,1]	81.10	90.22	89.56	88.91	90.22
[2,3,6,9]	[1,1,1,1]	81.08	90.56	89.55	88.49	90.65
[1,3,5,7]	[1,3,5,7]	81.13	90.39	89.58	88.78	90.39

constant dilation rates in the spatial branch help maintain geometric consistency of spatial features.

These findings provide empirical support for the adopted dilation strategy in the SESP module and highlight the importance of complementary dilation configurations across branches to optimize feature representation.

The detailed analysis and empirical evidence presented here enhance the understanding of dilation rate design and are incorporated in the revised manuscript (Section 2.3). To ensure reproducibility and facilitate future research, the full set of experiments has been made available in our public code repository.

7. Analysis on the Interaction and Efficiency of SESP, CESPB, and GETB in the Dual-Branch Framework

This supplementary material presents an in-depth analysis and additional experimental validation regarding the interaction among the SESP, CESPB, and GETB modules within our proposed dual-branch network architecture. Our network employs a dual-branch design, where the spatial branch primarily captures fine-grained spatial details, while the context branch focuses on modeling semantic dependencies and global contextual information. These two branches are integrated at multiple stages

to achieve complementary feature fusion.

However, excessive reliance on global context modeling—particularly when implemented solely via GETB modules in the context branch—can disrupt the balance between semantic richness and feature discriminability. This imbalance may introduce feature redundancy and reduce the diversity of learned representations, thereby negatively affecting segmentation performance.

To empirically investigate this, we conducted experiments comparing the combined SESP+CESPB+GETB design against a GETB-only configuration in the context branch. Specifically, the GETB-only setup increased the number of GETB modules from two to three while omitting the SESP and CESPB modules. This modification raised computational complexity from 9.206 GFLOPs and 1.661 MB parameters to 9.985 GFLOPs and 2.468 MB parameters. Contrary to expectations, this increase in resource consumption did not translate into accuracy gains; instead, key metrics including IoU and F1-score declined, as shown in Table 8. These results suggest that exclusive dependence on GETB modules for global context modeling leads to diminishing returns while incurring substantial computational overhead.

Table 8: Performance comparison between the combined SESP+CESPB+GETB modules and the GETB-only configuration in the context branch.

Method	IoU (%)	Acc (%)	F1-score (%)	Precision (%)	Recall (%)	Params (MB)	FLOPs (GB)
SESP+CESPB+GETB	81.46	90.13	89.78	89.43	90.13	1.661	9.206
GETBs	81.28	89.55	89.67	89.79	89.55	2.468	9.985

In addition, within the context branch, GETB modules are selectively applied at stages with relatively low channel dimensionality to balance efficiency and expressiveness. Replacing the entire context branch with GETB modules would necessitate applying them at higher channel dimensions, substantially increasing parameter count and FLOPs, thus compromising model efficiency—a critical factor for lightweight and mobile deployment.

The CESPB module's stage-wise cascading and adaptive dilation design achieves considerable receptive field enlargement at a modest computational cost, effectively capturing long-range dependencies while maintaining spatial structure. Ablation

studies further confirm that completely removing GETB modules results in noticeable accuracy degradation, indicating that global context modeling remains an indispensable component.

We also explored alternative architectural variants inspired by recent dual-branch frameworks, such as incorporating lightweight global aggregation modules at the end of the context branch (e.g., DDRNet's DAPPM and PIDNet's PAPPM). Although these modules have shown accuracy improvements in their original contexts, their integration into our network resulted in performance degradation, further emphasizing the necessity of balanced interplay between the spatial and context branches. Overemphasizing the context branch can overwhelm the spatial branch, impeding effective feature fusion and ultimately reducing segmentation accuracy.

In conclusion, the combined usage of SESP, CESPB, and GETB modules embodies a carefully balanced design, aimed at preserving fine spatial details, expanding receptive fields adaptively, and effectively modeling global semantic context. This analysis deepens our understanding of the complementary roles and efficiency trade-offs among these modules within the dual-branch architecture.

8. Distinct Design Principles and Feature Characteristics of the Spatial and Context Branches

This supplementary material provides a detailed clarification of the fundamental architectural and functional differences between the Spatial and Context branches in our dual-branch network. Although both branches receive the same input, their divergent design philosophies and processing strategies lead to markedly different feature representations.

The Spatial Branch is architected to preserve spatial resolution and channel dimensionality consistently throughout the network. Its primary objective is to maintain local structural information and fine-grained edge details. The integration of the SEAM module further strengthens the branch's ability to preserve boundary precision and spatial continuity.

In contrast, the Context Branch employs a progressive downsampling strategy while increasing the number of channels to extract high-level semantic context and

model global dependencies. This branch relies on the GETB module to effectively capture such global information.

Additionally, while both branches utilize the CESPB module, their dilation configurations differ substantially. The Spatial Branch uses a uniform dilation rate of [1, 1, 1, 1], which enables dense local detail extraction, whereas the Context Branch adopts progressively increasing dilation rates of [1, 2, 3, 4] to expand receptive fields and facilitate global feature learning.

These design distinctions—including differing receptive field structures and resolution/channel dynamics—result in complementary feature extraction by the two branches. This is visually confirmed through Grad-CAM-based feature visualizations presented in Figure 9, which compare the branches at multiple stages (low, middle, and high levels). The Spatial Branch predominantly emphasizes fine spatial structures and boundaries, while the Context Branch highlights high-level semantic understanding and global contextual relationships.

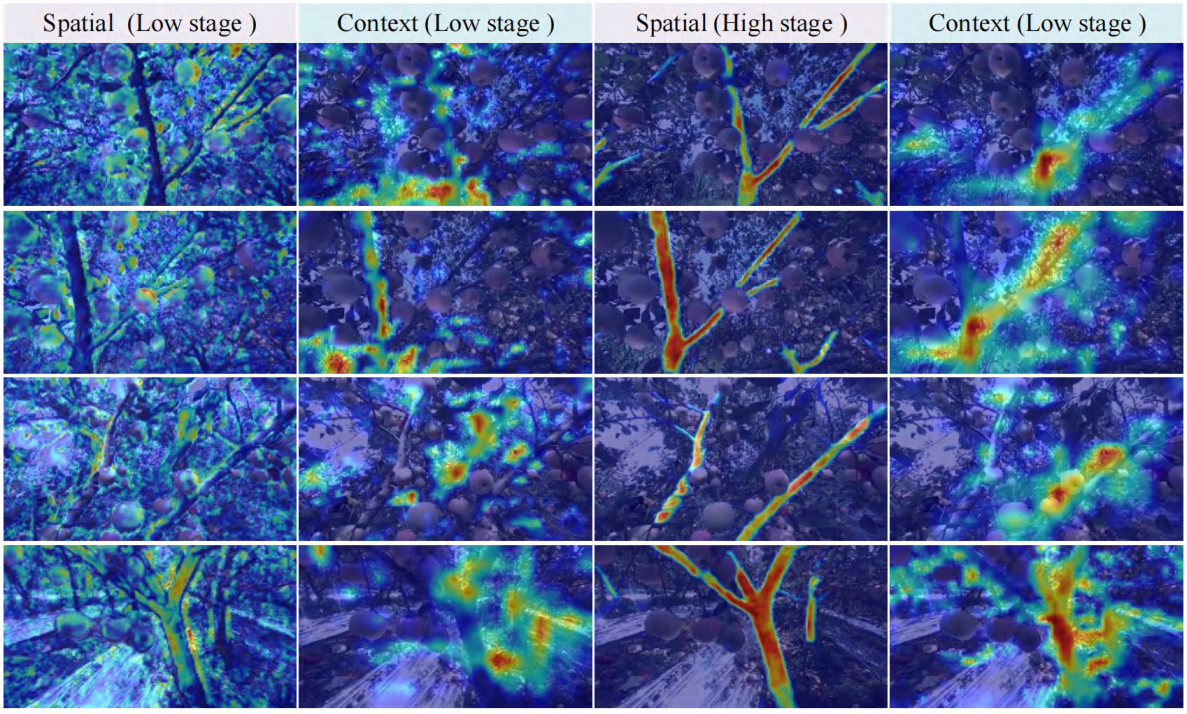

Fig. 9. Grad-CAM-based feature visualizations of the Spatial and Context branches at two different levels (low and high stages). The Spatial branch focuses on preserving fine-grained spatial structures and boundaries, while the Context branch emphasizes high-level semantic understanding and global dependencies.

This analysis clarifies the complementary roles of the two branches and reinforces the design rationale underlying the dual-branch architecture.