Quantization-Aware Super-Resolution for Real-Time Embodied Visual Perception

Anonymous Author(s)

***Abstract*—In an embodied intelligence system such as a mobile robot, visual perception on relatively low-quality images captured by onboard cameras must run in real time under limited computing resources and power constraints. Using super-resolution (SR) as an on-device perception front-end can improve visual quality for downstream perception tasks, but existing methods often fail to balance image quality, processing speed, and efficient deployment. In this work, we design a Dual-Cross Gate Network (DCGNet), as a hardware-aware SR architecture for real-time embodied visual perception. The DCGNet adopts a dual-stream structure that separately processes spatial and channel information, thus enabling efficient feature extraction. The network is trained with quantization-aware strategies and a lightweight attention mechanism to ensure robustness under robotic low-precision deployment. Then it is deployed on 8-bit integer arithmetic for real-world humanoid applications to test accuracy losses. Experimental results show that the DCGNet achieves real-time performance of 106 frames per second at 2K resolution, while maintaining image quality comparable to its full-precision version. These results verify that DCGNet can serve as a practical and deployable visual enhancement module for embodied perception under real-world system constraints.**

I. Introduction

In embodied intelligence systems such as mobile robots and service robots, visual perception must operate directly on-device under strict constraints of latency, power, and memory, while still supporting reliable downstream tasks such as navigation and manipulation [1], [2]. Due to motion blur, viewpoint jitter, and limited sensing quality, onboard cameras often capture visually degraded inputs, making super-resolution (SR) a practical means to enhance visual quality at the sensor level before higher-level perception modules [3]. However, unlike cloud-based or offline vision pipelines, SR in robotic systems must be deployed entirely on-device and operate in real time [1].

Most existing SR approaches, however, are primarily developed to maximize reconstruction accuracy in high-resource settings [4]. Representative methods typically adopt deep hierarchical convolutional backbones with wide channel dimensions, and increasingly rely on Transformer-based architectures to model global context and recover fine scale details [5]. While these designs have proven highly effective for visual reconstruction, they implicitly assume abundant computation and memory resources, making it challenging to deploy such models as real-time perception front-ends on resource-constrained robotic platforms [6].

So far, achieving real-time performance under such deployment constraints remains a major bottleneck for super-resolution in robotic systems. Unlike high-level perception tasks that allow feature downsampling, SR operates directly on high-resolution sensor data, leading to computational complexity that scales quadratically with input resolution. This poses a substantial burden on edge processors and makes many accuracy-oriented SR models impractical for real-time on-device deployment [7].

Conversely, edge deployment further imposes two major constraints. First, mobile Neural Processor Units (NPUs) are typically optimized for low-precision integer operations, making 8-bit quantization a necessary condition for compatibility and efficiency. Second, embedded systems have strict memory limits, and unoptimized SR architectures may cause frequent runtime failures due to memory overuse [8]. Similar optimization under practical resource and system constraints is also widely studied in other engineering domains [9].

Motivated by these observations, we propose DCGNet, an on-device SR front-end for real-time embodied perception. Our key insight is that efficient and quantization-robust reconstruction benefits from *separating* spatial-detail recovery from channel-wise feature modulation, while allowing *selective cross-interaction* between the two through lightweight gating. This decoupled-yet-cooperative design aligns the network's computation with the characteristics of mobile accelerators, and enables stable integer inference without sacrificing reconstruction fidelity. Fig. 1 illustrates the deployment of DCGNet within a robotic perception pipeline.

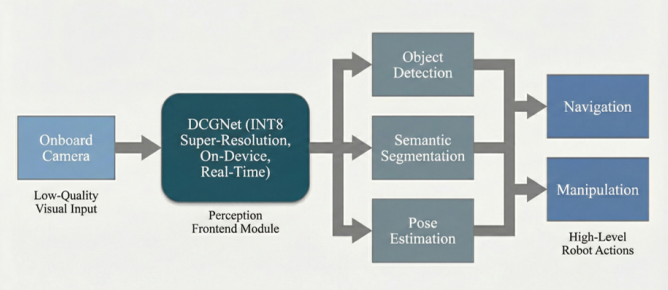

Fig. 1. System-level illustration of DCGNet as an on-device super-resolution front-end for embodied visual perception. DCGNet is deployed between the onboard camera and downstream perception modules (e.g., object detection, semantic segmentation, and pose estimation) to enhance visual quality directly on device, supporting high-level robot actions such as navigation and manipulation under real-time and resource-constrained settings.

The main contributions of this work are summarized as follows:

- We introduce a quantization-robust dual-stream architecture with cooperative feature fusion, enabling accuracy-

preserving INT8 super-resolution inference.
- By decoupling spatial reconstruction and channel modulation and incorporating quantization-aware training, we achieve deployment-ready real-time inference on edge devices.
- Experiments on Set5 [10], Set14 [10], BSD100 [11], and Urban100 [12] show that DCGNet achieves competitive SR quality with improved inference speed and is effective as an on-device front-end for downstream perception tasks.

II. PROPOSED METHODS

DCGNet is specifically designed for deployment on mobile devices and embodied platforms, where computational resources are significantly constrained compared with cloud-based systems.In such embodied perception systems, visual enhancement must be performed on-device and in real time to support downstream perception and decision-making.Therefore, each component of the architecture must be carefully optimized to reduce computational complexity while maintaining the required performance levels.

A. Framework Overview

As illustrated in Fig.2, DCGNet consists of five main components: the head block, the DSAF module, the enhanced SaE attention block, the EQCU modele, and the tail block.

The head block is responsible for extracting low-level features suitable for mobile hardware through lightweight convolutional reparameterization. DSAF module is composed of two structurally heterogeneous branches: one focusing on spatial texture extraction using depth-wise convolutions and the other emphasizing lightweight pattern analysis at the channel level. Their outputs are dynamically fused via a 1×1 convolution for cross-level recalibration. Let the network input be denoted as $I^{(in)}$. The outputs of the two asymmetric branches are represented as:

$$I^{(B_1)} = \mathcal{F}_{B_1}(\mathbf{H}(I^{(in)})), \quad I^{(B_2)} = \mathcal{F}_{B_2}(\mathbf{H}(I^{(in)})) \tag{1}$$

where $\mathcal{F}_{B_1}$ and $\mathcal{F}_{B_2}$ denote the convolutional operations in the spatial and channel branches, respectively. These outputs are fused through a parametric fusion mechanism:

$$I^{(fused)} = \text{DSAF}(I^{(B_1)}, I^{(B_2)}) = C_{1\times1}(I^{(B_1)} \oplus I^{(B_2)}) \tag{2}$$

Here, $\oplus$ denotes channel-wise concatenation and $C_{1\times1}$ is a learnable 1×1 convolution. The design intuition of DSAF is efficient feature decoupling and complementary fusion: the spatial branch emphasizes local texture and edge continuity, while the channel branch captures cross-channel dependency and global response patterns, so their fusion provides stronger reconstruction cues under limited computation. The SaE module is adopted and enhanced through the integration of SCConv, enabling fine-grained spatial-channel collaborative reconstruction. This mechanism recalibrates the fused features by aggregating multi-branch statistics in both spatial and frequency domains. Compared with conventional attention, the enhanced SaE is tailored for quantization-aware lightweight reconstruction: it keeps the calibration pathway simple and structured while preserving spatial–channel selectivity, which improves robustness to quantization noise and reduces deployment overhead on mobile hardware. To support quantized deployment, the EQCU module is introduced before the final output, optimizing feature pathways for integer inference through a reparameterized quadratic connection. The design intuition of EQCU is to improve deployment-oriented accuracy-efficiency trade-off by reusing intermediate features with a stable connection form, which benefits gradient propagation during training and enhances quantization robustness at inference. Finally, the tail block performs lightweight convolution and upsampling to generate the super-resolved image. The overall output of DCGNet is formulated as:

$$I^{(out)} = \mathbf{T}(\text{EQCU}(\text{SaE}(I^{(fused)}))) \tag{3}$$

The output of DCGNet is directly fed into downstream perception modules without requiring any modification to task-specific networks.

B. Optimization with Dynamic Asymmetric Quantization

To enable real-time deployment on resource-constrained and integer-only hardware platforms, we integrate dynamic asymmetric QAT into DCGNet. A key innovation is the use of channel-wise extremum tracking, which adaptively calibrates the dynamic range of convolutional activations. Specifically, the channel offset β_c is updated by an exponential moving average: Eq. (5) defines the dynamic-range calibration operation in the quantization module, where the offset state is adaptively updated before activation quantization.

$$\beta_c^{(t)} = \lambda_\beta \cdot \beta_c^{(t-1)} + (1 - \lambda_\beta) \cdot \text{EMA}(x_c^{(t)}), \tag{4}$$

where $x_c^{(t)}$ denotes the input activation of channel c at iteration t, $\text{EMA}(x_c^{(t)})$ is the running channel statistic estimated from the current mini-batch, $\beta_c^{(t-1)}$ and $\beta_c^{(t)}$ are the previous and updated channel offsets, and $\lambda_\beta = 0.9$ is the momentum coefficient. This operation takes channel activations and historical offsets as input and outputs the updated per-channel offset vector for asymmetric quantization in the QAT branch. Concretely, it first estimates the current channel statistic, then performs momentum-based weighted fusion with the historical offset, and finally outputs a stable offset to define quantization boundaries for subsequent feature reconstruction. This strategy ensures stable quantization boundaries and mitigates accuracy degradation during 8-bit inference. For multi-branch structures such as those in our enhanced SaE module, we adopt the standard practice of applying independent quantization during training and uniform quantization after branch fusion at inference, ensuring efficient deployment without additional complexity. Overall, the proposed quantization framework selectively applies asymmetric quantization while preserving critical attention modules, achieving hardware-efficient inference with minimal accuracy loss.

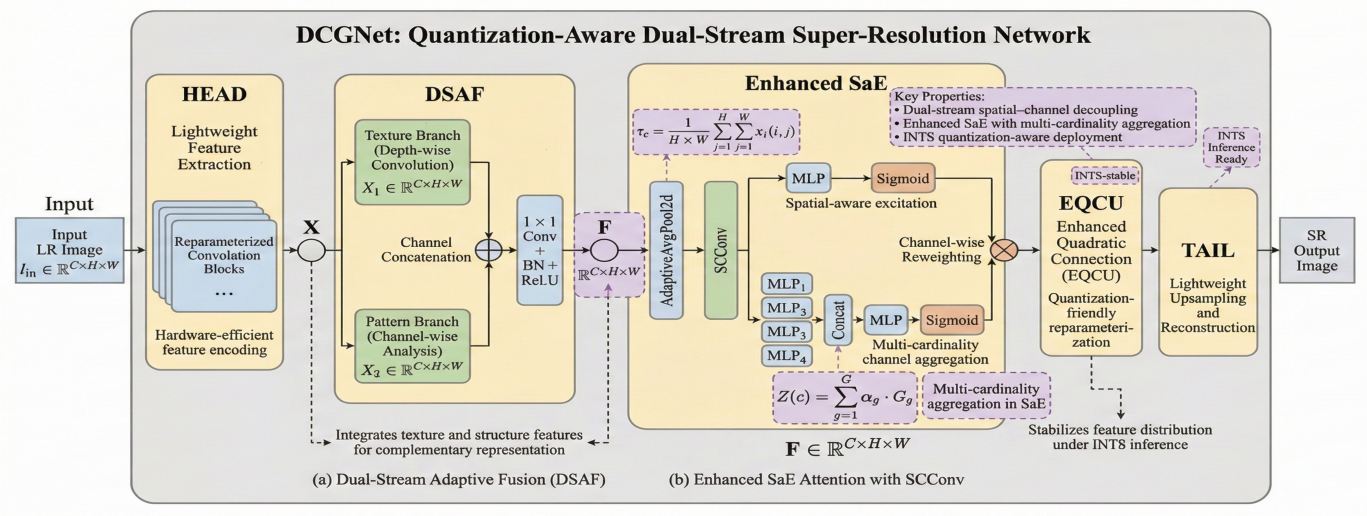

Fig. 2. Overall framework of DCGNet and key modules. The left part illustrates the full architecture, consisting of head, dual-stream adaptive fusion, enhanced squeeze aggregated excitation (SaE) mechanism, enhanced quadratic connection, and tail. The right part shows the internal designs of Enhanced SaE Attention (top) and DSAF (bottom): the SaE block integrates spatial and channel reconstruction convolution (SCConv) and multi-branch aggregation for refined spatial–channel calibration, while the DSAF module fuses heterogeneous branches via channel concatenation and a 1×1 convolution with BN+ReLU for efficient feature integration.

C. Structural Enhancements for Quantization Robustness

To ensure lossless accuracy under quantization, we introduce three structural enhancements. First, the enhanced SaE attention mechanism [13], enhanced by integrating SCConv [14] and multi-cardinality aggregation, significantly advances feature calibration for low-level vision tasks. During training, the SaE module introduces three key modifications: SCConv processing after global feature squeezing, multi-branch dynamic weight fusion, and inference-phase parameterization decoupling. Together, these enhancements improve feature refinement and enhance deployment efficiency. The fusion process can be formulated as:

$$Z^{(c)} = \sum_{g=1}^{G} \alpha_g \cdot G_g \tag{6}$$

The overall structure is illustrated in Fig. 2.

Second, the DSAF module establishes complementary heterogeneous branches with dynamic feature binding:

$$F = C_{1\times1}([B_1(x) \oplus B_2(x)]) \tag{8}$$

achieving nonlinear diversity, feature recalibration, and computational efficiency. The structure is shown in Fig. 2. Third, the enhanced outlier-aware loss function employs local statistics, EMA smoothing, and adaptive weighting, with the final form:

$$\mathcal{L} = \frac{1}{H \cdot W} \sum_{i,j} w_{i,j} \cdot \delta_{i,j}^2 \tag{13}$$

which is particularly beneficial for real-world embodied perception scenarios where visual inputs are often noisy and partially degraded.

III. EXPERIMENTS

A. Experimental Setup

Training Settings. All SR models are trained on the DIV2K dataset. During training, input images are randomly cropped into 256×256 patches and augmented with random horizontal and vertical flips as well as 90-degree rotations. The Adam optimizer is employed with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a cosine annealing learning rate schedule.

Evaluation Datasets. We evaluate SR performance on four widely adopted benchmark datasets: Set5, Set14, BSD100, and Urban100. These datasets cover a diverse range of natural scenes and urban structures and are commonly used for evaluating reconstruction fidelity.

Inference Platforms. Inference efficiency is evaluated on both GPU and CPU platforms. Quantized INT8 models are evaluated using the same inference pipeline to ensure fair comparison. Latency and throughput are measured under identical input resolutions.

Downstream Detection Setup. To evaluate the impact of SR as a perception front-end for embodied systems, we employ a pretrained YOLOv8n object detection model. All detection model parameters are frozen during evaluation to isolate the influence of different SR front-ends.

Implementation Details and Quantization Settings. Unless otherwise specified, all SR models are trained with $\times 2$ bicubic degradation on DIV2K for 600 epochs using a batch size of 16. Each mini-batch contains random 256×256 LR crops (corresponding to 512×512 HR regions), random horizontal/vertical flips, and 90° rotations. We use Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$, an initial learning rate of 2×10^{-4}, and cosine annealing to 1×10^{-6}.

TABLE I
COMPARISON ON SUPER-RESOLUTION PERFORMANCE BY PSNR AND SSIM WITH REPRESENTATIVE METHODS EVALUATED ON MOBILE GPU.

Method	Scale	#P	Avg latency (ms)	FPS (2K)	Set5	Set14	BSD100	Urban100
Accuracy-Oriented and Transformer-Based SR								
RDN [15]	$\times 2$	22.12M	1K+	< 1	38.24/0.9614	34.01/0.9212	32.34/0.9017	32.89/0.9353
RCAN [16]	$\times 2$	12.47M	1K+	< 1	38.27/0.9614	34.12/0.9216	32.41/0.9027	33.34/0.9384
HAN [17]	$\times 2$	64.61M	1K+	< 1	38.27/0.9614	34.16/0.9217	32.41/0.9027	33.35/0.9385
DRLN [18]	$\times 2$	34.43M	1K+	< 1	38.27/0.9616	34.28/0.9231	32.44/0.9028	33.37/0.9390
CISR [19]	$\times 2$	9.60M	1K+	< 1	28.94/0.8160	26.78/0.7080	26.08/0.6590	24.93/0.7270
Lightweight CNN-Based SR Methods								
SRCNN [20]	$\times 2$	19.6K	168	5	36.66/0.9542	32.42/0.9063	31.36/0.8879	29.50/0.8946
ABPN [21]	$\times 2$	33.5K	86.6	12	36.72/0.9556	32.49/0.9076	31.33/0.8891	29.39/0.8955
TPSR-D2 [22]	$\times 2$	60.8K	105	10	37.18/0.9578	32.84/0.9112	31.64/0.8935	30.24/0.9073
FSRCNN (D32S6M4) [23]	$\times 2$	5.78K	48.9	20	36.29/0.9510	32.20/0.9040	31.10/0.8840	28.91/0.8860
ESPCN (D64S32) [24]	$\times 2$	24.48K	54.8	18	36.64/0.9530	32.46/0.9070	31.32/0.8870	29.37/0.8930
SCNet-T [25]	$\times 2$	159K	10.2	101	37.85/0.9600	33.39/0.9161	32.06/0.8981	31.50/0.9187
ASID-SR [26]	$\times 2$	298K	21	63	38.17/0.9613	33.96/0.9216	32.31/0.9017	32.88/0.9353
SYENet [7]	$\times 2$	4.93K	16.5	60	36.84/0.9564	32.62/0.9079	31.52/0.8907	30.37/0.9029
Quantization-Aware and Deployment-Ready SR								
DCGNet-QAT (Ours)*	$\times 2$	5.22K	**9.6**	**106**	36.97/0.9525	34.01/0.9637	31.50/0.8867	30.27/0.9072

*DCGNet-QAT is evaluated under INT8 quantized inference.

TABLE II
IMPACT OF DIFFERENT SUPER-RESOLUTION FRONT-ENDS ON DOWNSTREAM OBJECT DETECTION PERFORMANCE UNDER EMBODIED VISUAL SETTINGS, EVALUATED WITH A FIXED DETECTOR AND IDENTICAL INFERENCE CONFIGURATIONS.

SR Front-End	**Detection Metric (mAP@0.5)**	**SR FPS**	**End-to-End FPS**
Bicubic	28.5	> 1000	47 (Detector limit)
SYENet	34.5	60	27 (SR bottleneck)
DCGNet-QAT	**34.4**	**106**	**34 (Real-time)**

Note: End-to-end FPS measures the overall throughput of the SR–detection pipeline.

For quantization-aware training, we first pretrain DCGNet in FP32 and then fine-tune with fake-quant operators for the last 100 epochs. Fake quantization is inserted on convolution weights and post-activation tensors in DSAF, SaE, and EQCU blocks. We adopt 8-bit quantization for both weights and activations, with per-channel symmetric quantization for weights and dynamic asymmetric per-channel quantization for activations (Eq. (5)). Observer statistics are initialized using 512 randomly sampled training patches and updated with EMA during QAT.

For deployment benchmarking, inference is conducted on a Snapdragon 8 Gen 2 mobile GPU and a desktop x86 CPU using the same runtime backend. Latency/FPS in Table I are reported with batch size 1 at 2K output resolution, averaged over 500 synchronized runs after 100 warm-up iterations. End-to-end FPS in Table II includes both SR and frozen YOLOv8n inference.

B. Comparison with Lightweight SR Methods

We compare DCGNet with representative super-resolution methods under the $\times 2$ upscaling setting, covering both accuracy-oriented models and lightweight designs. Quantitative results in terms of PSNR, SSIM, model size, inference latency, and throughput are summarized in Table I. All methods are evaluated under the same resolution setting on a mobile GPU, while models that are not feasible for real-time or on-device deployment are reported for reference only.

As shown in Table I, DCGNet-QAT achieves a favorable balance between reconstruction quality and deployment efficiency under strict lightweight constraints. With only **5.22K parameters**, DCGNet-QAT attains competitive reconstruction performance, achieving **36.97 dB / 0.9525 SSIM on Set5**, and maintaining strong results on more challenging datasets such as **Set14 (34.01 dB / 0.9637)** and **BSD100 (31.50 dB / 0.8867)**. Notably, these results are obtained with an inference latency of only **9.6 ms** and a throughput of **106 FPS at 2K resolution**, satisfying real-time requirements for on-device and embodied perception scenarios.

In contrast, many accuracy-oriented SR models, including recent Transformer-based approaches, achieve higher PSNR and SSIM at the cost of substantially increased model complexity and computational overhead, making them impractical for real-time deployment on resource-constrained platforms. Compared with recent lightweight CNN-based SR methods, DCGNet-QAT delivers comparable reconstruction quality while operating at lower parameter counts and significantly higher throughput. These results demonstrate that the proposed quantization-aware dual-stream design effectively preserves reconstruction fidelity under INT8 inference, while enabling

deployment-ready performance for real-time, on-device, and embodied visual perception systems.

C. Impact on Embodied Visual Perception

a) Experimental Setup: We adopt a pretrained YOLOv8n detector as the downstream perception model due to its widespread use in real-time robotic and embedded vision systems. YOLOv8n is selected as it represents a widely adopted lightweight detector whose performance is sensitive to input visual quality under real-time constraints. The detector is kept frozen during evaluation to ensure that any performance variation originates solely from the input quality rather than detector retraining. Detection is performed on low-resolution inputs processed by different SR front-ends.

Specifically, three input pipelines are evaluated: bicubic upsampling, a representative lightweight SR baseline, and the proposed DCGNet-QAT. All SR models operate under the same scaling factor and inference resolution, and the detection model processes the super-resolved images using identical settings.

b) Quantitative Results on Downstream Detection: Quantitative detection results are summarized in Table II, where performance is measured using standard detection metrics. As shown in the table, replacing conventional upsampling with learned SR consistently improves detection accuracy, confirming the importance of visual enhancement in embodied perception pipelines.

Table II shows that learned SR front-ends improve detection over bicubic (28.5 mAP@0.5). SYENet improves accuracy to 34.5 mAP@0.5 but bottlenecks end-to-end throughput at 27 FPS. DCGNet-QAT reaches 34.4 mAP@0.5 while sustaining 106 SR FPS and 34 end-to-end FPS, preserving real-time deployability. Qualitative examples in Figs. 3 and 4 further show clearer structures and more reliable pedestrian detection with DCGNet-QAT under INT8 inference.

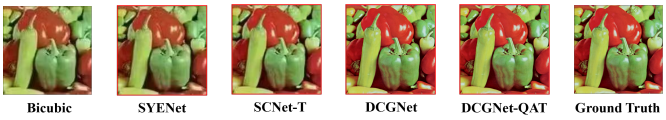

Fig. 3. Visual comparison on the Peppers image. DCGNet-QAT preserves visual quality close to high-accuracy baselines and remains nearly indistinguishable from its full-precision counterpart.

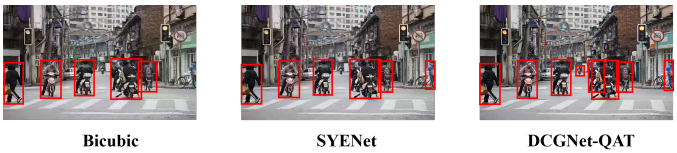

Fig. 4. Qualitative downstream detection comparison with bicubic, SYENet, and DCGNet-QAT. DCGNet-QAT preserves clearer structures under INT8 inference and enables more reliable pedestrian detection.

D. Quantitative Results and Analysis

Quantitative results on downstream object detection are summarized in Table II. As shown in the table, the choice of SR front-end has a significant impact on both detection accuracy and system-level throughput. While the pipeline in Fig.1 illustrates DCGNet as a general perception front-end applicable to multiple downstream tasks, we focus on object detection in this work as a representative embodied perception task.

All SR front-ends are evaluated using their deployment-ready configurations, and DCGNet is tested in its INT8 quantized form.

Table II indicates that the SR front-end strongly affects both downstream accuracy and system throughput. Bicubic upsampling gives the highest raw SR speed but the lowest detection accuracy (28.5 mAP@0.5), showing that insufficient visual detail limits downstream perception.

SYENet improves detection to 34.5 mAP@0.5 but reduces end-to-end throughput to 27 FPS because SR becomes the bottleneck. Under the same deployment-ready setting, INT8 DCGNet-QAT reaches 34.4 mAP@0.5 and restores higher system throughput (34 FPS), yielding the best accuracy–efficiency trade-off for this pipeline. In other words, DCGNet-QAT preserves detection-relevant details while keeping the overall embodied pipeline within a real-time operating range. These findings further suggest that SR front-ends for embodied perception should be evaluated not only by reconstruction quality, but also by their impact on end-to-end throughput and practical system-level evaluation protocols [27].

Importantly, DCGNet-QAT maintains real-time performance throughout the pipeline. The additional SR front-end introduces minimal overhead, and the combined SR–detection system continues to operate at frame rates suitable for embodied applications such as navigation and human–robot interaction. These results demonstrate that DCGNet enhances downstream perception not only in terms of accuracy but also in terms of system-level deployability.

IV. CONCLUSION

In this work, we have developed the DCGNet, a quantization-robust dual-stream super-resolution network designed as an on-device perception front-end for real-time embodied visual systems. Unlike conventional SR methods, which are primarily optimized for offline reconstruction accuracy, the DCGNet explicitly targets low-precision and resource-constrained deployment scenarios by centering the design around stable INT8 inference.

By decoupling spatial-detail reconstruction from channel-wise feature modulation and enabling lightweight cooperative interaction between the two streams, DCGNet preserves reconstruction fidelity under integer-only inference while maintaining high inference efficiency. Extensive experiments on standard SR benchmarks demonstrate that DCGNet achieves competitive reconstruction quality with significantly improved throughput, validating the effectiveness of the proposed quantization-aware design.

Further, we evaluate DCGNet in an embodied perception setting by integrating it into an end-to-end object detection pipeline. The results show that DCGNet improves downstream detection performance compared with conventional upsampling methods, while avoiding the computational bottlenecks introduced by heavier SR models. These findings indicate that super-resolution, when designed with quantization robustness in mind, can function as a practical and reliable perception front-end for real-time on-device embodied systems.

We also acknowledge a current limitation of this study: downstream validation is performed in one representative pipeline only, i.e., object detection with a frozen YOLOv8n detector. Extending the evaluation to additional detector architectures and broader embodied perception tasks will be an important direction of future work.

REFERENCES

[1] C. D. Singh, B. He, C. Fermüller, C. Metzler, and Y. Aloimonos, "Minimal perception: Enabling autonomy in resource-constrained robots," *Frontiers in Robotics and AI*, vol. 11, 2024. [Online]. Available: https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2024.1431826

[2] J. Luo, X. Zhou, C. Zeng, Y. Jiang *et al.*, "Robotics perception and control: Key technologies and applications," *Micromachines*, vol. 15, no. 4, p. 531, 2024. [Online]. Available: https://doi.org/10.3390/mi15040531

[3] H. Wang, R. Guo, P. Ma, C. Ruan, X. Luo, W. Ding, T. Zhong, J. Xu, Y. Liu, and X. Chen, "Event camera meets mobile embodied perception: Abstraction, algorithm, acceleration, application," *arXiv preprint arXiv:2503.22943*, 2025. [Online]. Available: https://arxiv.org/abs/2503.22943

[4] Z. Chen, E. Zamfir, N. Martinel, Z. Wu, and R. Timofte, "Ntire 2024 challenge on image super-resolution (×4): Methods and results," in *Proceedings of the NTIRE Workshop at IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPRW 2024)*. IEEE/CVF, 2024. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2024W/NTIRE/papers/Chen_NTIRE_2024_Challenge_on_Image_Super-Resolution_x4_Methods_and_Results_CVPRW_2024_paper.pdf

[5] J. Liu, J. Lin, W. Dong, X. Zhao, J. Liu, and H. Li, "Baid: A lightweight super-resolution network with binary attention-guided frequency-aware information distillation," *Computers, Materials & Continua*, vol. 86, no. 2, pp. 1–19, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1546221825012779

[6] A. Ignatov *et al.*, "Quantized image super-resolution on mobile npus: Mobile ai 2025 challenge," in *Proceedings of the Mobile AI 2025 Workshop at IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPRW 2025)*. IEEE/CVF, 2025. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2025W/MAI/papers/Ignatov_Quantized_Image_Super-Resolution_on_Mobile_NPUs_Mobile_AI_2025_Challenge_CVPRW_2025_paper.pdf

[7] W. Gou, Z. Yi, Y. Xiang, S. Li, Z. Liu, D. Kong, and K. Xu, "Syenet: A simple yet effective network for multiple low-level vision tasks with real-time performance on mobile device," *arXiv preprint arXiv:2308.08137*, 2023. [Online]. Available: https://arxiv.org/abs/2308.08137

[8] Y. LeCun, "1.1 deep learning hardware: Past, present, and future," in *2019 IEEE International Solid-State Circuits Conference - (ISSCC)*, 2019, pp. 12–19.

[9] J. Li, Z. Chen, S. He, A.-J. Li, and S. Yu, "Priority-based graph-enhanced reinforcement learning for robust analog circuit optimization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, no. 43, 2026, pp. 37 027–37 035.

[10] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International Conference on Curves and Surfaces*. Springer, 2010, pp. 711–730.

[11] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of the Eighth IEEE International Conference on Computer Vision (ICCV)*, vol. 2. IEEE, 2001, pp. 416–423.

[12] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5197–5206.

[13] N. Mahendran, "Senetv2: Aggregated dense layer for channelwise and global representations," *arXiv preprint arXiv:2311.10807*, 2023. [Online]. Available: https://arxiv.org/abs/2311.10807

[14] J. Li, Y. Wen, and L. He, "Scconv: Spatial and channel reconstruction convolution for feature redundancy," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 6153–6162.

[15] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2472–2481.

[16] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 286–301.

[17] B. Niu, W. Wen, W. Ren, X. Zhang, L. Yang, S. Wang, K. Zhang, X. Cao, and H. Shen, "Single image super-resolution via a holistic attention network," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 191–207.

[18] S. Anwar and N. Barnes, "Densely residual laplacian super-resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 3, pp. 1192–1204, 2022.

[19] A. Gunawan and S. R. H. Madjid, "CISRNet: Compressed Image Super-Resolution Network," arXiv preprint, 2022, arXiv:2201.06045.

[20] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 184–199.

[21] Z. Du, J. Liu, J. Tang, and G. Wu, "Anchor-based plain net for mobile image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2021, pp. 2494–2502.

[22] R. Lee, L. Dudziak, M. Abdelfattah, S. I. Venieris, H. Kim, H. Wen, and N. D. Lane, "Journey towards tiny perceptual super-resolution," in *European Conference on Computer Vision (ECCV)*, 2020.

[23] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," *CoRR*, vol. abs/1608.00367, 2016. [Online]. Available: https://arxiv.org/abs/1608.00367

[24] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1874–1883.

[25] G. Wu, J. Jiang, K. Jiang *et al.*, "Fully 1×1 convolutional network for lightweight image super-resolution," *Machine Intelligence Research*, vol. 21, pp. 1062–1076, 2024. [Online]. Available: https://doi.org/10.1007/s11633-024-1501-9

[26] K. Park, J. Soh, and N. I. Cho, "Efficient attention-sharing information distillation transformer for lightweight single image super-resolution," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 6, 2025, pp. 6416–6424.

[27] J. Li, H. Zhi, R. Lyu, W. Li, Z. Bi, K. Zhu, Y. Zeng, W. Shan, C. Yan, F. Yang *et al.*, "Analoggym: an open and practical testing suite for analog circuit synthesis," in *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, 2024, pp. 1–9.