Quantum CNN-GCN-Based Road Hazard Detection with Lyapunov Optimized Control

Juhui Heo, and Soohyun Park
Division of Computer Science, Sookmyung Women's University, Seoul, Republic of Korea
E-mail: heojuhui@sookmyung.ac.kr, soohyun.park@sookmyung.ac.kr

***Abstract*—While Cooperative Adaptive Cruise Control (CACC) enhances traffic efficiency by reducing inter-vehicle distances, shorter headways inevitably make it more challenging to respond promptly and reliably to unexpected road hazards. To address this issue, we assume a system in which Connected Vehicles (CVs) perform real-time hazard detection outside the CACC framework and disseminate the detection results to all vehicles through RSUs. To this end, we propose a lightweight model that integrates Graph Convolutional Networks (GCNs) and Quantum Convolutional Neural Networks (QCNNs), referred to as Quantum Graph-based Object Detection (QGOD), which exhibits a clear trade-off between detection accuracy and processing latency. Furthermore, the model incorporates a Lyapunov optimization-based control mechanism to dynamically adjust the model architecture according to traffic conditions, thereby maximizing the time-average performance.**

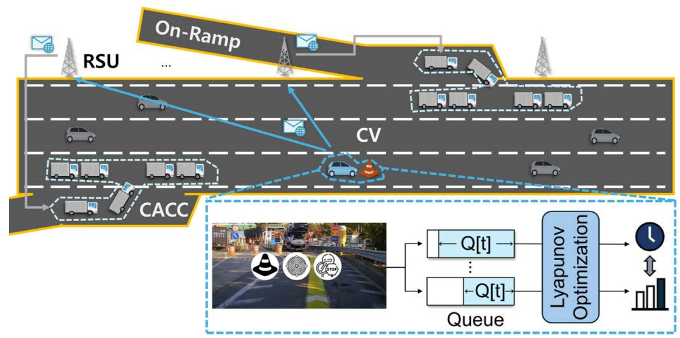

Fig. 1: Overall system architecture of the proposed QGOD framework.

I. Introduction

Cooperative Adaptive Cruise Control (CACC) is a key technology in intelligent transportation systems, enabling vehicles to maintain shorter headways and thereby improving overall traffic efficiency [1]. However, as inter-vehicle distances decrease, unexpected road hazards—such as manholes, construction signs, or debris—pose critical safety challenges. In such scenarios, the limited reaction time can result in delayed responses and potential accidents. To mitigate this risk, Connected Vehicles (CVs) operating ahead of CACC platoons play a crucial role by detecting road hazards in advance and transmitting this information to Roadside Units (RSUs). Through this process, the RSUs can promptly disseminate hazard information to CACC platoons, enabling timely and coordinated responses. Consequently, real-time road hazard detection and information sharing are indispensable for ensuring the safe operation of CACC systems [2]. Vision-based hazard detection has been widely studied, with Convolutional Neural Networks (CNNs) demonstrating superior accuracy and inference speed compared to sequential models such as Long Short-Term Memory [3]. Despite these advantages, CNN-based models typically require millions of parameters and substantial computational resources, which restricts their feasibility in autonomous vehicles (AVs) that must operate under strict onboard computing and latency constraints. Quantum Convolutional Neural Networks (QCNNs) have recently emerged as a promising alternative. By leveraging quantum parallelism and entanglement, QCNNs can achieve CNN-level performance with significantly fewer parameters [4]. This property makes them particularly suitable for resource-constrained and latency-sensitive AV environments. Similar to classical CNN-based models, where deeper architectures typically yield better performance, quantum models also demonstrate improved accuracy as the number of PQC layers increases; however, this inevitably leads to longer inference times [5].

To address these limitations, this study proposes a Quantum Graph-based Object Detection (QGOD) architecture that integrates QCNNs with Graph Convolutional Networks (GCNs). The key idea is to reduce the dimensionality of output features while maintaining detection performance, thereby preserving the efficiency of quantum feature extraction while mitigating computational complexity [6], [7]. This hybrid design combines the parameter efficiency of QCNNs with the relational learning capabilities of GCNs, providing a balanced solution for real-time perception. The overall system concept is illustrated in Figure 1. In this framework, Connected Vehicles (CVs) utilize the proposed QCNN-GCN model to detect road hazards such as potholes, construction signs, and debris. The detected information is then transmitted to RSUs and shared with CACC platoons, ensuring timely and reliable hazard awareness. Furthermore, the system incorporates Lyapunov-based optimization to guarantee stability and efficiency under heterogeneous vehicular computing constraints [8].

In summary, CNN-based models face significant computational and latency constraints that hinder their deployment in real-time vehicular systems. QCNNs, on the other hand, achieve parameter efficiency through quantum parallelism and entanglement, yet their computational burden increases as the output feature dimensionality grows, leading to higher inference latency. To overcome these challenges, this study proposes a QGOD architecture that reduces output feature dimensionality while preserving detection performance. By combining the lightweight encoding advantages of QCNNs with the relational reasoning capabilities of GCNs, the proposed model improves both training efficiency and inference speed, ultimately providing a balanced solution that bridges the gap between classical and quantum approaches in autonomous vehicle perception.

II. Related Work

A. CNN-based Road Hazard Detection

Deep learning models, particularly CNN-based detectors, have been widely applied in autonomous driving for tasks such as pothole and obstacle detection. Classical models like YOLO, Faster R-CNN, and Mask R-CNN demonstrate high accuracy and robustness in structured environments [9]. However, their deployment in AVs faces challenges due to high computational cost, memory usage, and latency constraints [10]. In AV systems, real-time perception is safety-critical, yet CNNs typically require millions of parameters and powerful GPU resources, which are not always feasible in embedded vehicular platforms [11].

B. Advances in QCNN

CNNs have been widely adopted in vision-based perception tasks for autonomous vehicles due to their high accuracy and fast inference capability [12]. Nevertheless, their reliance on millions of trainable parameters and high computational costs limits their real-time deployment on resource-constrained vehicular hardware [13]. To overcome these limitations, QCNNs replace classical convolution operations with quantum gates and measurements. By exploiting quantum parallelism and entanglement, QCNNs can achieve performance comparable to CNNs while using significantly fewer parameters [4], making them a promising alternative for lightweight and latency-sensitive applications in autonomous driving systems [14]. A typical quantum convolutional layer in QCNNs comprises four key components: (i) an encoder that transforms classical inputs into quantum states by rotating qubits on the Bloch sphere, (ii) CNOT gates that generate entanglement among qubits to capture spatial correlations across image patches, (iii) measurement in the Z-basis that collapses quantum states into classical real-valued outputs for subsequent processing, and (iv) a parameterized quantum circuit (PQC) layer consisting of RX, CNOT, and CU3 gates with trainable parameters applied to the CU3 gates.

III. Proposed Method

A. QGOD for Road Hazard Detection

A Quantum CNN Object Detection (QCOD) model employs a *Classical CNN* backbone to extract initial feature maps, passes them through the *Quantum Convolutional Layer*, and utilizes a *Detection Head* to predict objectness, bounding box coordinates, and class probabilities, forming a lightweight detection pipeline.

The QGOD model extends QCOD by incorporating a ***GCN* layer**, which aggregates global contextual information based on node connectivity, thereby capturing richer spatial relationships among objects in complex scenes and enhancing detection performance.

Fig. 2 illustrates the architectural overview of the proposed QGOD models.

1) Quantum Convolutional Layer: The QGOD model first employs a *Classical CNN* backbone to extract initial feature maps, which are subsequently fed into the *Quantum Convolutional Layer*. This layer partitions the input image into patches, encodes each patch into quantum states, and performs quantum operations. The primary quantum gates employed in this process are as follows:

- **Rotation gates** (R_x, R_y, R_z): Convert real-valued inputs, such as pixel intensities, into rotation angles for encoding into quantum states, thereby mapping continuous inputs into quantum representations.
- **CNOT gates**: Introduce entanglement between qubits, enabling the capture of local features within a patch, effectively serving as the quantum analogue of convolutional filters in classical CNNs.
- **CU3 gates**: Parameterized controlled rotation gates that perform data-dependent transformations, enhancing the ability to learn complex feature interactions.

By leveraging this combination of quantum gates, the *Quantum Convolutional Layer* achieves high representational capacity with significantly fewer parameters than classical convolution, while quantum entanglement allows the effective extraction of nonlinear relationships within each patch.

2) Graph Convolutional Layer: The feature maps extracted from the *Quantum Convolutional Layer* are processed by a post-activation block consisting of batch normalization and ReLU, followed by conversion into a grid-structured graph. Each grid cell is defined as a node, and edges are constructed based on four-neighborhood adjacency and self-loops. The channel value at each node corresponds to the feature dimension obtained after Quantum Convolutional Layer-based quantum operations, measurement, and a subsequent linear transformation.

The GCN aggregates information from neighboring nodes to expand local features into global contextual representations. The update rule of the GCN is expressed as

$$h_v^{(l+1)} = \sigma\left(\sum_{u\in\mathcal{N}(v)\cup\{v\}} \hat{A}_{vu} W^{(l)} h_u^{(l)}\right), \tag{1}$$

where $h_v^{(l)}$ denotes the embedding of node v at the l-th layer, $\hat{A}$ is the normalized adjacency matrix with self-loops, $W^{(l)}$ is a learnable weight matrix, and σ is a nonlinear activation function. This operation preserves local spatial structures while effectively incorporating global information, and it can also be applied to multi-resolution feature maps for learning contextual information at different scales.

3) Classical Head: The multi-resolution feature maps enhanced by the GCN are finally passed to the *Classical Head*. At each grid location, predefined anchors with various scales and aspect ratios are generated, and the detection head simultaneously predicts objectness scores, bounding box

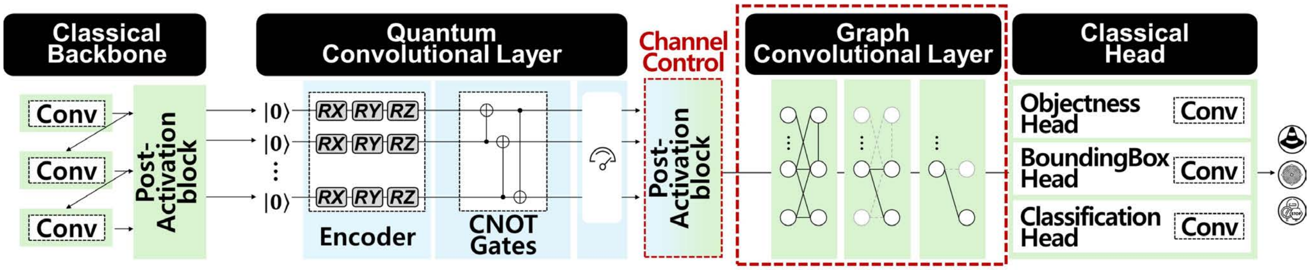

Fig. 2: Architectural overview of the QGOD models.

coordinates, and class probabilities. During training, IoU-based matching is employed to assign positive and negative samples. The final outputs consist of objectness scores, location offsets, and class predictions, enabling real-time object detection in autonomous driving scenarios.

B. Lyapunov Control for Real-Time Detection

1) Queue Modeling for QGOD Processing: To model the delay problem in the QGOD framework, we consider a discrete-time queueing system where $Q[t]$ denotes the queue length at time t, $a[t]$ represents the number of new data samples arriving at time t, and $b(\alpha[t])$ denotes the processing rate determined by the selected PQC-depth $\alpha[t]$. The queue dynamics are governed by

$$Q[t+1] = \max(Q[t] - b(\alpha[t]), 0) + a[t], \tag{2}$$

where

- $a[t]$: arrival process, i.e., number of images arriving at time t,
- $b(\alpha[t])$: departure process, i.e., number of images processed under the chosen PQC-depth $\alpha[t]$.

A larger queue length $Q[t]$ corresponds to longer latency, whereas a smaller $Q[t]$ ensures real-time responsiveness. Therefore, maintaining *queue stability*, i.e., keeping the time-average queue length bounded, is crucial for preventing excessive delays in real-time autonomous driving systems.

2) Time-average QGOD Model Control: To balance detection accuracy and queue stability, we employ the *Lyapunov Drift-Plus-Penalty* optimization framework. The objective is to maximize the time-average detection accuracy while maintaining queue stability, formulated as

$$\max: \lim_{T\to\infty} \sum_{t=0}^{T-1} A(\alpha[t]) \tag{3}$$

$$\text{s.t.} \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} Q[t] < \infty. \tag{4}$$

The Lyapunov function is defined as

$$L(Q[t]) = \frac{1}{2}(Q[t])^2, \tag{5}$$

with the conditional Lyapunov drift

$$\mathbb{E}[L(Q[t+1]) - L(Q[t])|Q[t]]. \tag{6}$$

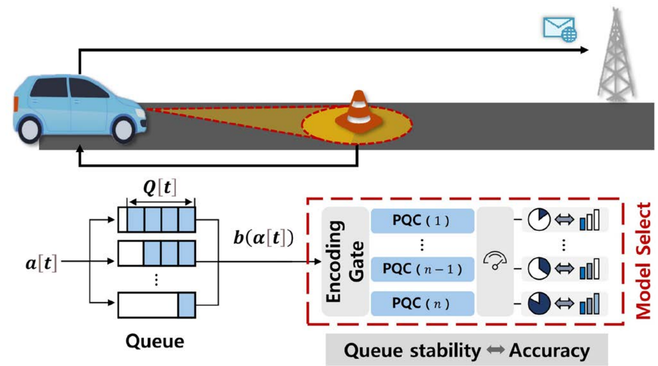

Fig. 3: Adaptive PQC-depth selection for real-time QGOD operation via Lyapunov optimization.

The drift upper bound can be derived as

$$\Delta(Q[t]) \le C + \mathbb{E}[Q[t](a[t] - b(\alpha[t]))|Q[t]], \tag{7}$$

where C is a constant satisfying

$$\frac{1}{2}\mathbb{E}[a[t]^2 + b(\alpha[t])^2|Q[t]] \le C. \tag{8}$$

Since neither C nor $a[t]$ can be controlled, the drift-plus-penalty objective is defined as

$$\alpha^*[t] \leftarrow \arg\max_{\alpha[t]\in\mathcal{A}} \left[V \cdot A(\alpha[t]) - Q[t] \cdot b(\alpha[t])\right], \tag{9}$$

where $V > 0$ controls the trade-off between detection accuracy and delay. A smaller number of PQC layers is selected under large $Q[t]$ to reduce latency, while a larger number of PQC layers is applied when $Q[t]$ is small to improve accuracy. This adaptive PQC-depth allocation balances real-time responsiveness with detection performance.

IV. Performance Evaluation

In this study, the proposed models were evaluated on the *Road Hazard Detection* task for Connected Vehicles (CVs). The dataset employed was developed by KISTI for object detection applications and consists of GoPro videos recorded on major roads in Daejeon, South Korea, during July-September 2019 and July-September 2020 using dedicated vehicles. Each frame was annotated with polygonal object labels as the primary annotation format, along with bounding box information. Furthermore, privacy-sensitive regions, such as pedestrians, faces, and license plates, were anonymized through blurring.

(a) Manhole

(b) Construction signs & Parking prohibited board

(c) Traffic cone

Fig. 4: Sample images from the dataset representing dif road conditions.

TABLE I: Performance and Complexity Comparison of C cal CNN, QCOD, and QGOD Models

Model	mAP@0.5 (%)	Number of Parameters	Inference Time (sec)
classical CNN	32.16	190, 280	0.97×10^{-2}
QCOD(16 channels)	13.56	39, 894	1.33×10^{-2}
QCOD(32 channels)	34.10	42, 134	1.45×10^{-2}
QGOD(16 channels)	33.49	40, 198	1.21×10^{-2}

For the experiments, three classes were selected: *Construction signs & Parking prohibited board*, *Traffic cone*, and *Manhole*, comprising a total of 190 images. Among these, 148 images were used for training and 42 images for testing. All images were converted to grayscale, resized to a resolution of 90×160 (H×W), and pixel intensities were normalized to enhance training stability and consistency. To compensate for the limited number of samples, data augmentation techniques such as rotation, flipping, cropping, and noise injection were applied to enhance the diversity of the training dataset. Example images for each class are provided in Fig. 4.

Table I summarizes the quantitative comparison of detection performance (mAP@0.5), number of parameters, and inference time among Classical CNN, QCOD, and QGOD models. The classical CNN, even when achieving performance comparable to the quantum models (mAP@0.5 = 32.16%), requires 190,280 parameters and an inference time of 0.97×10^{-2} seconds, which highlights its drawback of higher computational complexity. For the QCOD model, performance and complexity vary with the number of channels. The QCOD model with 16 channels is the most lightweight configuration with only 39,894 parameters, but it yields the lowest mAP@0.5 (13.56%). Increasing the channel count to 32 improves the mAP@0.5 to 34.10% while requiring 42,134 parameters and increasing inference time to 1.45×10^{-2} seconds. In contrast, the QGOD (16 channels) model achieves comparable detection accuracy (mAP@0.5 = 33.49%) with 40,198 parameters and 1.21×10^{-2} seconds of inference time, demonstrating a more efficient trade-off between model complexity and performance compared to QCOD.

TABLE II: Test Accuracy and Frames Per Second (FPS) by Number of PQCs

PQC Layers	Accuracy (%)	Inference Time (sec)
1	33.49	1.21×10^{-2}
2	38.18	1.90×10^{-2}
3	44.87	2.43×10^{-2}

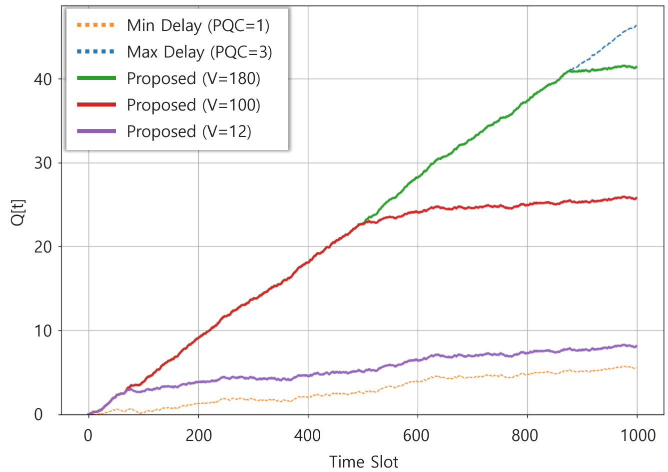

Fig. 5: Queue backlog dynamics in the Lyapunov-optimized QGOD model under varying V values. Larger V values place greater emphasis on model accuracy, resulting in a higher queue backlog.

To enable real-time inference, we developed a QCNN model that dynamically adjusts the number of PQC layers via Lyapunov optimization. Table II presents the performance and inference latency with respect to the number of PQC layers. As shown, increasing the number of PQC layers improves detection accuracy: the test accuracy rises from 33.49% with one PQC layer to 44.87% with three PQC layers. This improvement is attributed to the enhanced quantum feature extraction and entanglement representation enabled by deeper PQC architectures. However, higher accuracy comes at the cost of increased computational latency. For instance, the single-layer model infers one image in 1.21×10^{-2} seconds, whereas the three-layer model requires 2.43×10^{-2} seconds.

The proposed system employs Lyapunov optimization to adaptively balance detection accuracy and inference latency by adjusting the PQC depth according to real-time system states. Instead of relying on a fixed QGOD architecture, the Lyapunov controller selects model configurations that ensure queue stability while achieving the target detection performance.

Fig. 5 illustrates the effect of varying the Lyapunov control parameter V on queue backlog. Larger V values prioritize accuracy, leading to the selection of deeper QCNN models and, consequently, higher queue backlogs. Conversely, smaller V values favor shallower models with lower latency. In this study, $V = 12, 100, 180$ are tested. When V is too small, the queue is under-utilized, resulting in degraded performance, whereas excessively large values cause queue divergence, making them unsuitable for real-time deployment.

Compared to fixed single-PQC models, the proposed approach achieves a more effective trade-off between accuracy and latency. These results demonstrate that Lyapunov-based control provides a flexible and dynamic framework for optimizing real-time inference in connected vehicle applications.

V. Conclusion and Future Work

In this study, we proposed a novel road hazard detection model that integrates QCOD with GCNs, referred to as the QGOD architecture, which provides three major contributions. First, even under accuracy levels comparable to quantum models, conventional CNN-based models still require a large number of parameters, making them unsuitable for deployment in real-time and resource-constrained environments such as autonomous vehicles and edge computing platforms. Second, the QCOD models exhibit a clear trade-off between detection accuracy and computational efficiency depending on their architectural configurations. In contrast, the proposed QGOD model incorporates graph structural information into the quantum convolutional framework, achieving comparable accuracy while significantly reducing both the number of parameters and inference latency. This demonstrates that leveraging graph connectivity can enhance representational capacity without increasing computational complexity. Finally, the Lyapunov-based PQC-depth optimization dynamically adjusts model depth according to real-time system conditions, effectively balancing detection accuracy and inference latency. As a result, the proposed approach enables adaptive detection systems capable of meeting both real-time and efficiency requirements across diverse traffic scenarios.

For future work, we plan to extend the proposed model into a CV-RSU-based content management framework, integrating reinforcement learning policies to establish an end-to-end perception-decision-making pipeline that encompasses content generation, storage, and transmission. Although the introduction of GCN layers has successfully reduced the number of trainable parameters, the detection performance remains unsatisfactory. To address this issue, we will strengthen empirical validation using larger datasets and state-of-the-art baselines, while further refining the model architecture to improve both accuracy and efficiency. Furthermore, to support practical CACC operations, we will develop an enhanced CV-RSU communication architecture capable of real-time transmission of hazard detection information, and conduct an in-depth discussion on the feasibility of practical quantum hardware implementation, thereby bridging the gap between theoretical modeling and real-world deployment.

Acknowledgment

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-RS-2024-00436887) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

References

[1] J. Hu, S. Sun, J. Lai, S. Wang, Z. Chen, and T. Liu, "CACC Simulation Platform Designed for Urban Scenes," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 4, pp. 2857–2874, April 2023.

[2] W. Shao, Z. Fan, C.-J. Chen, Z. Zhang, J. Ma, and J. Zhang, "Impact of Sensing Errors on Headway Design: From αα-Fair Group Safety to Traffic Throughput," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 10 952–10 965, December 2024.

[3] A. S. Alhanaf, M. Farsadi, and H. H. Balik, "Fault Detection and Classification in Ring Power System With DG Penetration Using Hybrid CNN-LSTM," *IEEE Access*, vol. 12, pp. 59 953–59 975, April 2024.

[4] E. J. Roh, H. Baek, D. Kim, and J. Kim, "Fast Quantum Convolutional Neural Networks for Low-Complexity Object Detection in Autonomous Driving Applications," *IEEE Transactions on Mobile Computing*, vol. 24, no. 2, pp. 1031–1042, February 2025.

[5] K. Phalak, J. Li, and S. Ghosh, "Trainable PQC-Based QRAM for Quantum Storage," *IEEE Access*, vol. 11, pp. 51 892–51 899, May 2023.

[6] Z. Shan, Q. Yang, R. Ye, Y. Zhang, Y. Xu, X. Xu, and S. Liu, "GPA-Net:No-Reference Point Cloud Quality Assessment With Multi-Task Graph Convolutional Network," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 8, pp. 4955–4967, June 2024.

[7] P. Liang, L. Xu, H. Shuai, X. Yuan, B. Wang, and L. Zhang, "Semisupervised Subdomain Adaptation Graph Convolutional Network for Fault Transfer Diagnosis of Rotating Machinery Under Time-Varying Speeds," *IEEE/ASME Transactions on Mechatronics*, vol. 29, no. 1, pp. 730–741, July 2024.

[8] H. Baek, R. Lee, S. Jung, J. Kim, and S. Park, "Real-Time High-Quality Visualization for Volumetric Contents Rendering: A Lyapunov Optimization Framework," *IEEE Open Journal of the Computer Society*, vol. 4, pp. 243–252, September 2023.

[9] Y. Xiang, "Intelligent Fabric Pattern Detection and Design Based on Mask R-CNN," in *Proc. 2024 IEEE 7th International Conference on Information Systems and Computer Aided Education (ICISCAE)*, Dalian, China, September 2024, pp. 657–661.

[10] Z. Hua, K. Aranganadin, C.-C. Yeh, X. Hai, C.-Y. Huang, T.-C. Leung, H.-Y. Hsu, Y.-C. Lan, and M.-C. Lin, "A Benchmark Review of YOLO Algorithm Developments for Object Detection," *IEEE Access*, vol. 13, pp. 123 515–123 545, July 2025.

[11] D. Yang and Z. Luo, "A Parallel Processing CNN Accelerator on Embedded Devices Based on Optimized MobileNet," *IEEE Internet of Things Journal*, vol. 10, no. 21, pp. 18 844–18 852, May 2023.

[12] D. Chun, J. Choi, H.-J. Lee, and H. Kim, "CP-CNN: Computational Parallelization of CNN-Based Object Detectors in Heterogeneous Embedded Systems for Autonomous Driving," *IEEE Access*, vol. 11, pp. 52 812–52 823, May 2023.

[13] M. Cheshfar, M. H. Maghami, P. Amiri, H. G. Garakani, and L. Lavagno, "Comparative Survey of Embedded System Implementations of Convolutional Neural Networks in Autonomous Cars Applications," *IEEE Access*, vol. 12, pp. 182 410–182 437, December 2024.

[14] A. Rattan, A. Rudra Pal, and M. Gurusamy, "Quantum Computing for Advanced Driver Assistance Systems and Autonomous Vehicles: A Review," *IEEE Access*, vol. 13, pp. 17 554–17 582, January 2025.