# Application of lightweight target recognition models in edge computing

[1]Wei Jinchuan , [2]Zhou Wei
The 28th Research Institute of China Electronics Technology Group Corporation

## ABSTRACT

Edge computing environments place high demands on the design of target recognition models, requiring efficiency, real-time performance, and resource sensitivity. Lightweight target recognition models, with their advantages of low computational overhead and flexible deployment, are gradually becoming a core technology for edge intelligent applications. This paper focuses on the implementation mechanism, optimization path, and typical applications of lightweight models in edge computing, aiming to explore the application value of model compression, architecture optimization, and collaborative computing strategies in real-world scenarios, and to promote the deep integration of target recognition and edge intelligence.

**Keywords:** Lightweight Model; Target Recognition; Edge Computing

## 1. INTRODUCTION

With the widespread adoption of smart terminals and the surge in the number of IoT devices, an increasing number of computing tasks are being offloaded from the cloud to edge devices to improve response efficiency and protect data privacy. Object recognition, a crucial task in computer vision, plays a key role in scenarios such as intelligent surveillance, autonomous driving, and robot navigation. The deployment of traditional large-scale convolutional neural networks on edge devices is constrained by computing resources and energy consumption, necessitating lightweight model design and optimization strategies.

### I. Technical Basis and Design Strategy of Lightweight Target Recognition Model

#### (a) Performance constraints of edge computing on target recognition models

The target recognition model running in the edge computing environment needs to complete the inference task within the limited hardware resources and strict energy consumption budget. Compared with traditional cloud computing, edge devices usually have less memory, lower computing power and tighter power consumption requirements, which imposes strict restrictions on the parameter scale and computational complexity of the model [1]. The target recognition model needs to have efficient spatial feature extraction capabilities and maintain a high recognition accuracy in order to achieve stable operation on the edge device. The latency sensitivity in edge computing requires the model to complete the inference feedback in a very short time to avoid the degradation of system performance due to computational delay.

#### (ii) Lightweight model compression technology improves computational efficiency

Lightweighting of target recognition models usually relies on model compression techniques to reduce computational load by reducing redundant parameters and simplifying the structure. Model compression methods include strategies such as pruning, quantization, and knowledge distillation, which can significantly reduce the network size while maintaining core recognition performance. Pruning techniques remove unimportant weights or neurons from the model by analyzing the network , thereby reducing computational burden and speeding up inference. Quantization techniques convert model parameters from high-precision floating-point numbers to low-bit-width integers to reduce memory usage and accelerate arithmetic operations [2]. Knowledge distillation transfers the "knowledge" of complex networks to smaller networks, enabling lightweight networks to inherit the expressive power of large models. These compression strategies are used in conjunction in edge computing tasks, which helps to build efficient target recognition models that meet resource constraints.

#### (III) Efficient architecture design drives the evolution of lightweight models

For resource-constrained edge computing platforms, effective network architecture design is an important factor in improving the performance of lightweight target recognition. When designing a lightweight architecture, modularization and depthwise separable convolution are usually adopted to reduce the number of parameters and computational redundancy [3]. Depthwise separable convolution reduces computational complexity without significantly losing expressive power by decomposing standard convolution into two steps: channel-wise convolution and pointwise convolution. Modular structure can split different functional units, enabling the model to maintain high flexibility when performing specific tasks and supporting dynamic

adjustment of computational paths. Effective architecture design can also combine skip connections, multi-scale feature fusion and other technologies to improve the overall expressive power, enabling lightweight models to complete high-precision target recognition tasks on edge devices. These design strategies achieve a balance between lightweighting and performance of the model through network structure optimization, providing robust recognition capabilities for edge application scenarios.

## II. Application Practice and Optimization Path of Lightweight Target Recognition Model in Edge Computing

### (I) Real-time target detection solution on embedded edge devices

Deploying a target recognition system on an embedded edge device requires consideration of various hardware resource constraints, such as limited storage space, low computing power processing units, and strict energy consumption budgets, which poses a challenge to real-time recognition. Lightweight target recognition models enable neural networks to run on these limited platforms through structural optimization and compression strategies, enabling edge devices to quickly recognize targets in dynamic scenes. Real-time target detection in embedded environments usually combines lightweight models with hardware acceleration modules to reduce processing latency and increase frame rate, enabling the system to maintain high-frequency tracking of targets [4]. For example, in traffic monitoring scenarios, lightweight recognition models deployed on roadside units can detect targets such as vehicles and pedestrians in real time and feed the results back to the control center to optimize signal scheduling. Combined with edge computing resources, this scheme avoids dependence on the cloud, shortens response latency, and ensures data privacy through local inference, which provides reliable technical support for large-scale distributed intelligent monitoring and automatic decision-making systems.

### (II) Optimization of intelligent recognition services under the collaborative edge-cloud system

The application of lightweight target recognition models in edge computing is not an isolated deployment. It often collaborates with cloud platforms to form an edge-cloud hybrid architecture, balancing real-time performance with the processing capabilities for complex inference tasks. In this architecture, the edge is responsible for rapid inference and initial result generation of the lightweight model, while the cloud handles more complex deep analysis, long-term learning, and model updates. The rapid recognition results from the edge can serve as triggers to upload data requiring higher semantic understanding or cross-temporal correlation to the cloud for deep inference or storage analysis. This collaborative mechanism allows the lightweight model and the high-performance computing resources in the cloud to complement each other, effectively sharing the overall system load and accelerating response times. The cloud can also adjust the structure and weights of the lightweight model based on the data distribution at the edge, periodically pushing optimized model parameters to ensure the edge model continuously adapts to dynamically changing scenarios, improving the robustness and applicability of the overall intelligent recognition system.

### (III) Lightweight Model Fast Adaptation Strategy Based on Transfer Learning

Faced with the diversity of edge computing application scenarios and the non-stationary nature of data distribution, lightweight target recognition models need to possess rapid transfer and adaptation capabilities, achieving efficient deployment and performance improvement in new scenarios through transfer learning. Transfer learning strategies typically employ pre-trained models to learn basic features on large-scale datasets, then transfer the learned knowledge to specific edge scenarios. Only minor adjustments to certain layers or lightweight modules are needed to adapt to the data distribution of the target scenario. This rapid adaptation method reduces model training time, alleviates the pressure on edge device computing resources, and enables the model to maintain high recognition accuracy in different physical environments. For example, in the field of industrial inspection, adapting a general target recognition model to a specific product feature recognition task through transfer learning enables real-time detection and classification of product defects, improving the efficiency and stability of automated production lines. The introduction of transfer learning enhances the versatility and deployment flexibility of lightweight models in edge computing systems, enabling them to better cope with changing application requirements.

### (iv) Distributed optimization methods to enhance collaborative reasoning at the edge

In edge computing environments, multiple collaborative nodes may work together to complete target recognition tasks. To improve overall recognition efficiency and resource utilization, distributed optimization methods become crucial. Distributed collaborative inference can break down large recognition tasks into several sub-tasks, which are then executed collaboratively across multiple edge nodes. Parallel processing is achieved through task sharding and scheduling mechanisms, thereby reducing inference latency. Lightweight models can be dynamically allocated based on the computing power of each node in distributed collaboration, ensuring full utilization of edge computing resources. In complex environments, collaborative inference mechanisms also need to consider data transmission overhead and communication latency between nodes, balancing computational load by optimizing scheduling strategies and reducing cross-node data exchange. In traffic perception systems, distributed target recognition allows roadside equipment and camera nodes to share lightweight model inference results, enabling cross-view tracking of vehicles and pedestrians. This distributed collaborative approach enhances the recognition

system's ability to handle large-scale scenes, providing an efficient and stable implementation path for edge intelligence systems.

## 6. Conclusion

In conclusion, the application of lightweight target recognition models in edge computing platforms provides an efficient and low-energy solution for real-time intelligent services. By combining compression technology, collaborative computing architecture, transfer learning, and distributed optimization strategies, high-quality recognition performance can be achieved in resource-constrained environments. In the future, with the improvement of computing power in edge devices and the development of communication technologies, the application of such models will become more widespread, continuously providing technical support for fields such as intelligent sensing systems, intelligent transportation, and unmanned systems.

## REFERENCES

[1] Han Haiqing. Optimization application of lightweight neural network model in edge computing device [J]. Journal of Software, 2023, 43(02):78-87.

[2] Shen Shuying. Research on design and implementation of target recognition system supported by edge computing [J]. Computer Applications Research, 2023, (10): 112-118.

[3] Chen Suihai . Deep learning model compression technology and its application in visual recognition tasks [J]. Journal of Electronics, 2023, (08): 45-52.

[4] Wang Yuanhao, Zhang Guo, Wang Huachuan . IQN algorithm combined with priority experience replay mechanism [J]. Command Information Systems and Technology, 2025, 16(3):44-50.