Proceedings of the 4th International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/2026.35510

Lightweight Model Inference Techniques in Edge Computing Environments: Review

Hengrui Bi

Faculty of Science and Technology, University of Macau, Macau, China
dc42828@um.edu.mo

Abstract. Computation methods of artificial intelligence are gradually shifting from cloud computing to edge computing and on-device machine learning (ODML). How to contribute an effective machine learning model in the resource-limited environment, has become a significant and rapidly evolving research field. The training and inference of deep learning model used to be performed on the cloud high-performance computing clusters. There are many problems with uploading data to cloud, for example, high latency, round-trip latency, security issues, and a lack of privacy guarantees, and in this case, people cannot make real-time decisions. So, using edge devices to process tasks can significantly decrease the cost of transmission. The need of low latency, quick response, privacy protection and high adaptability has become the drive force of this change. This report aims to provide a comprehensive overview of lightweight model inference technologies in edge computing environments, mainly targeting low performance devices, such as mobile phones, intelligent equipment in vehicle, VR/AR headsets and Internet of Things (IoT). This paper introduces efficient learning and inference on edge devices from four aspects: 1) the definition of core terminology and concrete application environment; 2) the core technology of model compression, neural networks and knowledge distillation, which is used to deal with the tasks in the resource-limited environment; 3) the standards for evaluation of time/space complexity; 4) the challenge and opportunity which people face currently and in future.

Keywords: Lightweight model inference, Edge computing, Model compression, Lightweight neural networks, On-Device machine learning, Internet of things

1. Introduction

Early AI model highly depended on large cloud computing cluster. It means that all the data had to be uploaded to servers for processing. However, with the rapidly spreading of smart device, this way not only suffers from high latency and internet depend but also faced risk of leaking private data. In recent years, the computing ways have a critical shift from cloud AI to local computing (edge computing, ODML [1]). The driving forces behind deploying local offline model come from three key aspects. The first one is the strict requirement of low-latency, which is very important to latency sensitive application, such as autonomous driving, real-time rendering and the cooperation in operating factory lines. The following is the aspect about improving the protection of pravity. The risk of data leaking of cloud-based processing [2] can be solved by processing data locally. It can

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.35510

effectively protect the personal and official security. Finally, improving the environment adaption of model, locally processing models can effectively run in special environment which without internet connection, ensuring the availability when the connection is bad or even completely unconnected.

2. Scope of this study

2.1. Core terminology

For a long time, researchers make an effort to make AI services closer to end users and data source. Over time, edge computing [3-5] and on-device have developed many mature research frameworks and established a large number of significant concepts and ideas. ODML means putting model inference or lightweight training tasks directly onto smart devices, which enable the computation to be performed locally and make the protection of private data from the source come true. The dependence on cloud connection can also be decreased by ODML. Edge computing and edge AI are closely integrated with ODML, forming a structure in system level. It aims to use distributed computing model to provide real-time analyzing and deciding services which is near the edge nodes. Furthermore, Tiny ML [6] has emerged as a new cutting-edge field when facing the massive and resource-limited basic sensing nodes. This field is mainly related to how to use model compression; adaption between hardware and software and how to enable complex intelligent algorithms to run efficiently and continuously on ultra-low power microcontrollers (MCUs) and edge sensors.

2.2. Typical application scenarios

Currently, the number of smart terminals and IoT [7] devices are growing quickly. They directly generated large amounts of data, and prompted the application scenarios of edge computing and model inference. Most application scenarios target to heterogeneous or low-power devices, because these devices not only have requirements for latency and security, but also have limitations on energy consumption. The most mature application scenarios of edge inference appear on mobile devices. Smart phone can run some local AI assistant which uses real-time image processing technology. For example, optimizing exposure or filters automatically when taking photos, and make photos in mobile album to be clearer. Local processing not only can accelerate the interaction speed between users and devices, but also can prevent some user private data from being uploaded to the cloud. It is well mentioned that edge AI is equally important in vehicle devices and intelligent transporting system. Autonomous driving typically relies on local hardware to do the tasks about detecting the targets like pedestrians, obstacles or traffic light; predicting the tracks of pedestrians and cars; and enabling the intelligent voice interaction in vehicles. To ensure the safety of passengers, the time required to make key decisions must lower than milliseconds. It makes sure that, even if the vehicle in the poor internet connection area, local processing can still work fine [8]. Wearable and immersive devices are another key aspect that need to be concerned. As the main immersive devices, VR/AR need strong local computing ability for real-time 3D rendering and low latency eye tracking to prevent users from dizziness. Smartwatches and other wearable devices need to collect and process the bio signals of users in daily life and warning when the health reading is abnormal [5]. Edge models are also widely used in the larger IoT ecosystem [7]. They are in smart home control nodes and industrial production line sensors, which mainly used for detecting anomalies and performing maintenance. Running computations where data is initially generated can reduce bandwidth requirements and release the pressure of massive data of IoT uploads on cloud infrastructure.

DOI: 10.54254/2755-2721/2026.35510

3. Core technologies in resource-constrained environments

3.1. Model compression technology

Model compression [9] uses algorithms to shrink model size and reduce computation, but do not change the initial network structure. It is the main method for running model on edge devices. Network pruning is the most common compression technique [9]. First, it evaluates the importance of each parameter, and then removes redundant synaptic connections and neurons from the model to achieve model accuracy while compressing the model, just like a gardener trims the branches. It can immediately reduce the number of model parameters and the theoretical computation complexity, and make it become more suitable for embedded systems that are extremely sensitive to storage space, which is also the main advantage of this technology. It is also used to solve the problems of memory shortage in edge-end models. However, there are still some apparent disadvantages as well. Pruning techniques are divided into unstructured and structured. Unstructured pruning usually generates a sparse matrix filled with invalid data, and this sparse matrix cannot be computed by using ordinary GPUs or CPUs, which requires special hardware to accelerate the computation. Although structured pruning is hardware-friendly, it will lead to some degree of accuracy loss [10]. Another core technique is model quantization, which is commonly used in industry. In this technique, the usage of memory is significantly decreased by mapping high-precision floating-point parameters to low-precision integers [11]. This approach greatly compresses the memory usage, and make full use of edge chip-level integer arithmetic units to accelerate inference. Therefore, as a standard technology to achieve real-time edge detecting and voice woke-up, it is widely used in smart phones, smart watches and other devices. It mainly includes three aspects: linear quantization, extreme quantization, and mixed-precision quantization [11]. Linear quantization compresses numerical precision according to a fixed uniform rule. Extreme quantization saves storages by storing data with as much binary bits as possible. Mixed-precision quantization combines these two methods to find the balance point which is most suitable to the special model. There is no doubt that model quantization also has some disadvantages, because reducing the bit width can lead to quantization errors, and extremely low-bit quantization can even lead to precipitous decreasing in model performance.

3.2. Lightweight neural networks

Lightweight network design works differently from model compression. It builds microarchitecture to balance computing efficiency and model expressive ability, and the biggest challenge is finding the method about how to extraction feature performance. Because it has to support complex tasks such as computer vision but the computing resources are limited. It is widely used in drone visual navigation and smart home security. Currently, the main architectures include MobileNet, which introduces the concept of separable convolution, decomposing standard convolution into spatial dimension filtering and channel dimension fusion. The advantages of this decision is an exponential reduction in multiplication and addition operations, extremely improves execution speed. However, the limitation of it is the upper limit of model capacity, which making it prone to accuracy when handling highly complex classification tasks [12]. To solve the problem of obstructed information flow between channels caused by grouped convolution, ShuffleNet proposed a channel shuffling mechanism, which is effectively breaking down information barriers between computational groups and improving the robustness of its feature. However, in some memory-limited or bandwidth-limited hardware [13], frequently reading and writing in memory and data rearrangement operations may

DOI: 10.54254/2755-2721/2026.35510

offset the benefits of improving the computation. Furthermore, GhostNet can be used, while a large number of similar feature maps in deep networks. In GhostNet, model uses low-cost linear operations to generate repeated fake feature maps. In this method, the amount of feature maps can be increased with lowest cost, but it is very sensitive to the design and fine-tuning of network hyperparameters, which means the parameter of system need to be set by people, and subsequent tuning and optimization are very difficult [14].

3.3. Knowledge distillation

Knowledge distillation mainly uses classic teacher-student structure [15]. It based on soft labels or feature matching to create loss function. Its operating is like a teacher teaching student. A large teacher model learns implicit knowledge from hard-labeled data, and the output soft label knowledge from teacher model can be smoothly transferred to a smaller distilled student model. The significant advantage of this technology is enhancing performance with high efficiency even on low-performance hardware. Even without increasing the actual inference burden of small models, it can till significantly improve accuracy and generalization capabilities. This characteristic perfectly solves the problem that edge devices cannot directly run large models because of insufficient computing power or network congestion. Therefore, the teacher-student architecture is widely used in scenarios which requiring high accuracy and limited by latency, such as real-time voice interaction and intelligent driving systems in vehicle. Although, this technology is effective, it also faces many challenges. The training process is complexed, high performance teacher model needs a large number of data and pre-training, while the adjustment of the distillation loss function parameters heavily depends on research experience.

3.4. Hardware awareness and privacy protection mechanisms

Deploying models on resource-limited hardware requires consideration in many aspects. For one, the decision needs to adapt to the limitation of basic hardware. For another, it is necessary to protect the private and security of users. Hardware awareness network structure is built for solving this problem. It directly puts the latency and energy consumption of the target heterogeneous hardware as hard constraints into the searching loop. It will put the key limitation like running time, electric consumption into the model process of finding the best solution, and algorithm can find out the most suitable neural network structure. This method is mainly aimed at customized edge AI chips and can achieve the best balance between running speed and recognition accuracy, breaking through the limitations of designing hardware-software cooperation solutions by people. However, this approach has many limitations, mainly because its extremely high computational cost during the searching process and the poor ability of the generated models across different hardware platforms. In the aspect of private protection, as a mainstream solution for addressing the privacy needs of enterprises and individuals, federated learning allows models to be trained locally on edge devices, with only encrypted model parameter updates uploaded to the cloud for aggregating the parameters of all devices. This mechanism ensures that all data is restored locally, eliminating privacy risks at the source. It is particularly suitable for the scenario that requires protecting data privacy and involves multi-collaboration, such as joint screening for a certain disease by medical institutions or joint risk assessment by financial institutions [16]. It can also be combined with blockchain technology to make it safer [2]. The main shortage of this technique is that devices may be limited by the quality of the network connection, which can significantly slow down the training efficiency, when various

DOI: 10.54254/2755-2721/2026.35510

terminal devices transmit data to the cloud. Besides that, if the data coverage from different node devices are not same at all or without rules, the model will be difficult to coverage.

4. Efficiency evaluation standard

The inference efficiency of lightweight edge models can be evaluated from multiple aspects, including model accuracy, theoretical computational cost, hardware deployment costs, and actual consumption of operation. Task performance and accuracy are the fundamental standards for measuring model's core utility. Key indicators include classification accuracy, mean Average Precision (mAP) in detecting object, and the ability for crossing the data sets. These indicators can reflect the model's precision in different tasks and the adaptability and reliability of it in different application environment. The mAP is used to measure the ability to accurately locate the target and ensure correct classification. Generalization ability reflects the model's accuracy in predicting samples which is outside the training sets, and represents the ability of adaptation. At the aspect of theoretical complexity and model size, parameters such as Multiply-Accumulate Operations (MAC), Floating Point Operations (FLOPs), parameter numbers in model and model file size are usually used as the standard to evaluate the theoretical computing cost and the usage of storage space. These indicators are not limited by hardware platforms and can be compared between models directly. For the hardware usage and long-term costs of model deployment, parameters such as chip size, manufacturing cost, and inference consumption are typically used for evaluation. These indicators have important reference value for resource-limited edge devices such as battery powered devices and IoT nodes. Besides that, the runtime performance and resource utilization of the model are also key evaluation metrics. Random access memory (RAM) usage, flash memory usage, and external memory access frequency are also included. They can directly reflect the consumption of basic hardware resources by the model during inference, and they are core evaluation parameters during the stage of model deployment.

5. Future opportunities and challenges

Edge computing and lightweight inference show great potential in many application scenarios. However, applying it to actual production scenarios still faces many unsolved problems, and the lack of adaptive data is critical. At present, there are very few high-quality datasets available for simulating hardware behavior or algorithms on heterogeneous edge devices. This gap seriously limits the performance which can be achieved through hardware-software co-optimization. Besides that, strict physical limitation constitutes the obstacles of hardware. The high frequency operation on chips is not only limited by extremely limited memory bandwidth, but also a great challenge to the cool system of devices because of the high energy consumption and substantial heat generated. It forces people to find a balance point between extreme lightweight design and system thermodynamic stability. Compared with large cloud models which rely on massive parameters and abundant computational resource, lightweight models on edge devices still have significant shortages currently. While addressing complex environmental noise, unseen scenarios, and adversarial perturbations, its anti-interference robustness and adaptability exhibit significant limitations.

Besides these challenges, edge AI currently presents unprecedented growth potential. The underlying hardware is evolving at great paces. Semiconductor manufacturing processes continue to advance, and the performance of edge-specific AI accelerator chips improves in these years. These improvements break the physical limit of traditional mobile hardware, provide greater computing

DOI: 10.54254/2755-2721/2026.35510

foundation. In terms of algorithm evolution, with the increasing maturity of theoretical frameworks such as knowledge distillation and structured compression, the feature extraction efficiency of lightweight micro networks has a significant achievement. Their accuracy in specific tasks is continuously approaching to traditional large cloud models. What more important is the increasing requirement from businesses and individual users for privacy protection and localized data processing, and it introduces related techniques from laboratory to large-scale commercial deployment. This security requirement to prevent data from uploading to the cloud from the source provides extremely strong market force for edge computing and ODML.

6. Conclusion

Lightweight edge model inference is still in a tiny market. Even so, this technology still has greatly potential growth in future. This report systematically reviews the key advancements in edge computing, efficiency optimization, and machine learning. In addition, the report explores how can existing technologies work better when deployed on resource-limited end devices. Even these technologies face many challenges, the trends of AI shifting from cloud to device are unstoppable. To achieve this goal, great efforts are required from both academia and industry to reduce energy consumption, improve robustness and private protection.

References

[1] Dhar, S., Guo, J., Liu, J., Tripathi, S., Kurup, U., & Shah, M. (2021). A survey of on-device machine learning: An algorithms and learning theory perspective. *ACM Transactions on Internet of Things, 2*(3).
[2] Xu, Y., Mao, Y., Li, S., Li, J., & Chen, X. (2023). Privacy-preserving federal learning chain for internet of things. *IEEE Internet of Things Journal, 10*, 18364–18374.
[3] Andriulo, F. C., Fiore, M., Mongiello, M., Traversa, E., & Zizzo, V. (2024). Edge computing and cloud computing for internet of things: A review. *Informatics, 11*, 71.
[4] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal, 3*, 637–646.
[5] Ahmed, E., & Rehmani, M. H. (2017). Mobile edge computing: Opportunities, solutions, and challenges. *Future Generation Computer Systems, 70*, 59–63.
[6] Dutta, L., & Bharali, S. (2021). Tinyml meets iot: A comprehensive survey. *Internet of Things, 16*, 100461.
[7] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems, 29*, 1645–1660.
[8] Goudarzi, S., Soleymani, S. A., Anisi, M. H., Jindal, A., & Xiao, P. (2025). Optimizing uav-assisted vehicular edge computing with age of information: An sac-based solution. *IEEE Internet of Things Journal, 12*, 4555–4569.
[9] Reed, R. (1993). Pruning algorithms - a survey. *IEEE Transactions on Neural Networks, 4*, 740–747.
[10] Chen, X., Zhu, J., Jiang, J., & Tsui, C.-Y. (2023). Tight compression: Compressing cnn through fine-grained pruning and weight permutation for efficient implementation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 42*, 644–657.
[11] Wu, J., Leng, C., Wang, Y., Hu, Q., & Cheng, J. (2016). Quantized convolutional neural networks for mobile devices. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4820–4828.
[12] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4510–4520.
[13] Zhang, X., Zhou, X., Lin, M., & Sun, R. (2018). ShuffleNet: An extremely efficient convolutional neural network for mobile devices. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6848–6856.
[14] Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., & Xu, C. (2020). GhostNet: More features from cheap operations. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1577–1586.
[15] Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge distillation: A survey. *International Journal of Computer Vision, 129*, 1789–1819.
[16] Khan, L.U., Saad, W., Han, Z., Hossain, E. and Hong, C.S. (2021) Federated Learning for Internet of Things: Recent Advances, Taxonomy, and Open Challenges. *IEEE Communications Surveys & Tutorials*, 23, 1759-1799.