Software Orchestrated and Hardware Accelerated Artificial Intelligence: Toward Low Latency Edge Computing

Cailian Deng, Xuming Fang, *Senior Member, IEEE,* Xianbin Wang, *Fellow, IEEE,* and Kevin Law

Abstract

Driven by the expeditious wireless evolution and growing complexity of Internet of Things systems, edge intelligence has been widely recognized as a novel paradigm to enable ubiquitous smart industry and consumer applications, which uses mobile edge computing (MEC) to push artificial intelligence (AI) related computing to the network edge near mobile terminals and data sources for low-latency data processing. However, design and deployment of edge intelligence remain extremely challenging due to resource-constrained edge computing environments and latency-related considerations. These motivate us to explore edge intelligence from different perspectives, particularly software orchestration (e.g., AI model design, model optimization, and resource management) and hardware acceleration methods (e.g., AI chip customization). However, the performance improvement of edge intelligence using just one single perspective is extremely limited, especially when its advantages are exhausted, which seriously hinders further performance development. In this article, we propose an edge computing framework with co-acceleration of software orchestration and hardware to break through the bottleneck of a single acceleration perspective on performance improvement, enabling high-quality edge intelligence service deployment. Simulation results show that the proposed edge computing framework has dramatically improved system performance in facilitating low-latency edge intelligence.

Cailian Deng and Xuming Fang are with the Key Laboratory of Information Coding and Transmission, Southwest Jiaotong University, Chengdu 611756, China. E-mail: dengcailian@my.swjtu.edu.cn; xmfang@swjtu.edu.cn.

Xianbin Wang is with the Department of Electrical and Computer Engineering, Western University, London, ON N6A 5B9, Canada. E-mail: xianbin.wang@uwo.ca.

Kevin Law is with Huawei Tech., Shenzhen 518129, China. E-mail: k.law@huawei.com.

(Corresponding author: Xuming Fang.)

Index Terms

Artificial intelligence (AI), edge intelligence, edge computing, acceleration, AI chip.

I. Introduction

The recent advancement of artificial intelligence (AI) technologies will create a new paradigm shift from Internet of Things (IoT) to Internet of Intelligence in next-generation wireless communications. Predictably, AI technologies (e.g., machine learning and deep learning) will gradually replace parts of human roles and even will outperform humans in various fields with high work intensity, difficulty or dangerous environment, such as facial recognition, computer vision, natural language processing, and anomaly detection, improving work efficiency dramatically. The implementation of AI applications often involves two computing steps: training an AI model based on a large-scale training dataset and utilizing the trained AI model for data processing and decision-making (i.e., AI inference). Since these two steps usually require powerful computing power and massive storage resources, it is difficult to directly enable computation-intensive and latency-sensitive AI applications on resource-constrained mobile terminals. Therefore, AI training and inference so far mostly resort to the remote cloud datacenter (i.e., cloud-based centralized intelligence). Specifically, the raw data are transmitted to the cloud datacenter from mobile terminals on the network edge for complicated processing, and then the processing results are transmitted back to mobile terminals. In recent years, driven by the fast evolution of IoT and wireless communication technologies, the number of mobile terminals and the amount of data at the network edge have increased dramatically, making data sources undergo a radical shift from the cloud datacenter to the increasingly widespread mobile terminals. Unfortunately, processing large amounts of data generated by mobile terminals in the cloud datacenter often incurs unexpected end-to-end latency, high energy consumption, and privacy leakage issues.

To significantly improve AI processing performance, edge intelligence, which uses edge computing technology to push AI from the cloud datacenter to the network edge which is closer to mobile terminals and data sources, is getting wide attention [1] [2]. However, edge intelligence is still in its infancy and faces a fundamental challenge, that is, how to effectively build the entire training and inference process of AI models on the network edge. This challenge has motivated the research on software orchestration methods, including model designs, model optimizations, and computing framework designs. Model compression [3] has been commonly adopted to reduce the model complexity and accelerate the model inference, making AI models easier to deploy to mobile terminals or the network edge. Considering differentiated

computation capabilities among mobile terminals, edge servers and complicated network environments, a terminal-edge collaborative computing framework was proposed in [4], enabling on-demand low-latency inference by jointly designing the model partition strategy and the early-exit strategy. In [5], a computing-power networking framework was presented for ubiquitous AI by establishing networking in the AI computing-power pool, enabling the adaptability for computing-power users, the flexibility for networking, and the profitability for computing-power providers.

Besides the innovation of theoretical algorithms, edge intelligence is also inseparable from the most advanced chips' performance improvement. Although some lightweight AI models can run on general-purpose chips like central processing units (CPUs), it is tough to perform deeper deep neural networks on the CPUs due to huge intolerable training/inference latency and energy consumption. Therefore, many researches focus on designing specialized AI chips to significantly accelerate AI-related computing in an energy-efficient way, such as improving the level of parallel computing and reducing the frequency of memory access [6].

However, acceleration methods from only one perspective, such as hardware acceleration or software orchestration, gradually reach the computing performance bottleneck when the single perspective's advantages are exhausted. Therefore, to further improve AI computing performance, new effective solutions are expected from multiple perspectives (i.e., co-acceleration of software orchestration and hardware). Software orchestration focuses on AI model designs, model optimizations, computing framework designs, and effective resource management, whereas hardware acceleration focuses on hardware design.

In this article, to further promote the process of edge intelligence, we propose a low-latency edge computing framework to obtain better computing performance through co-acceleration of software orchestration and hardware. This framework can break through the performance bottleneck from a single acceleration perspective. Besides, we present a case study of edge computing with co-acceleration of software orchestration and hardware. Numerical results show the potential benefits of co-acceleration of software orchestration and hardware.

The rest of this article is organized as follows. In the following section, we propose an edge AI computing framework with co-acceleration of software orchestration and hardware for latency-sensitive AI applications. Then we present a case study for the proposed framework and show simulation results to verify the proposed scheme's effectiveness. Finally, we conclude the article.

II. Proposed Edge AI Computing Framework with Co-acceleration of Software Orchestration and Hardware

In this section, we propose an edge AI computing framework with multifaceted acceleration methods to help AI execute more efficiently and quickly, enabling low-latency edge intelligence. As shown in Fig. 1, this framework can be broadly characterized into the following three parts: AI Applications, Software Orchestration, and Hardware Acceleration.

A. AI Applications

Due to the remarkable advantages in analyzing large amounts of data and extracting features, AI has been widely applied in various fields, including Smart Home, Intelligent Manufacturing, Intelligent Transportation, and Intelligent Agriculture. Running computation-intensive and latency-sensitive AI applications on the network edge is highly resource-intensive and requires fast data processing by high-end chips. However, limited resources on mobile terminals and relatively slow evolution in the semiconductor industry are hindering AI applications' expansion. Therefore, there is an urgent need to further accelerate the training and inference of deep neural networks through hardware acceleration and software orchestration. With the co-acceleration of software orchestration and hardware, we will anticipate an unprecedented era where innovations are achieved cooperatively.

B. Hardware Acceleration

Chips for hardware acceleration have tremendous implications for applying AI to domains under significant constraints such as size, weight and power, both in the mobile terminals and edge network. AI chips mainly include graphics processing units (GPUs), field-programmable gate arrays (FPGAs), and application-specific integrated circuits (ASICs) highly specialized for AI applications [6]. Other types of chips, such as general-purpose CPUs, can also be used for simple or lightweight AI tasks, but CPUs are becoming tougher as AI advances. Table 1 compares the typical AI chips and general-purpose chips.

1) CPUs: The CPU has two key components, that is, the control unit (CTRL) in charge of handling all operation instructions and the arithmetic logic units (ALUs) used to perform mathematical calculations. The CPU is capable of both management and computing. However, the CPU has no specific computing and memory budget designed for AI algorithms. If taking AI tasks, it will result in extremely low energy efficiency. Normally, the CPU is only affordable for lightweight AI computation.

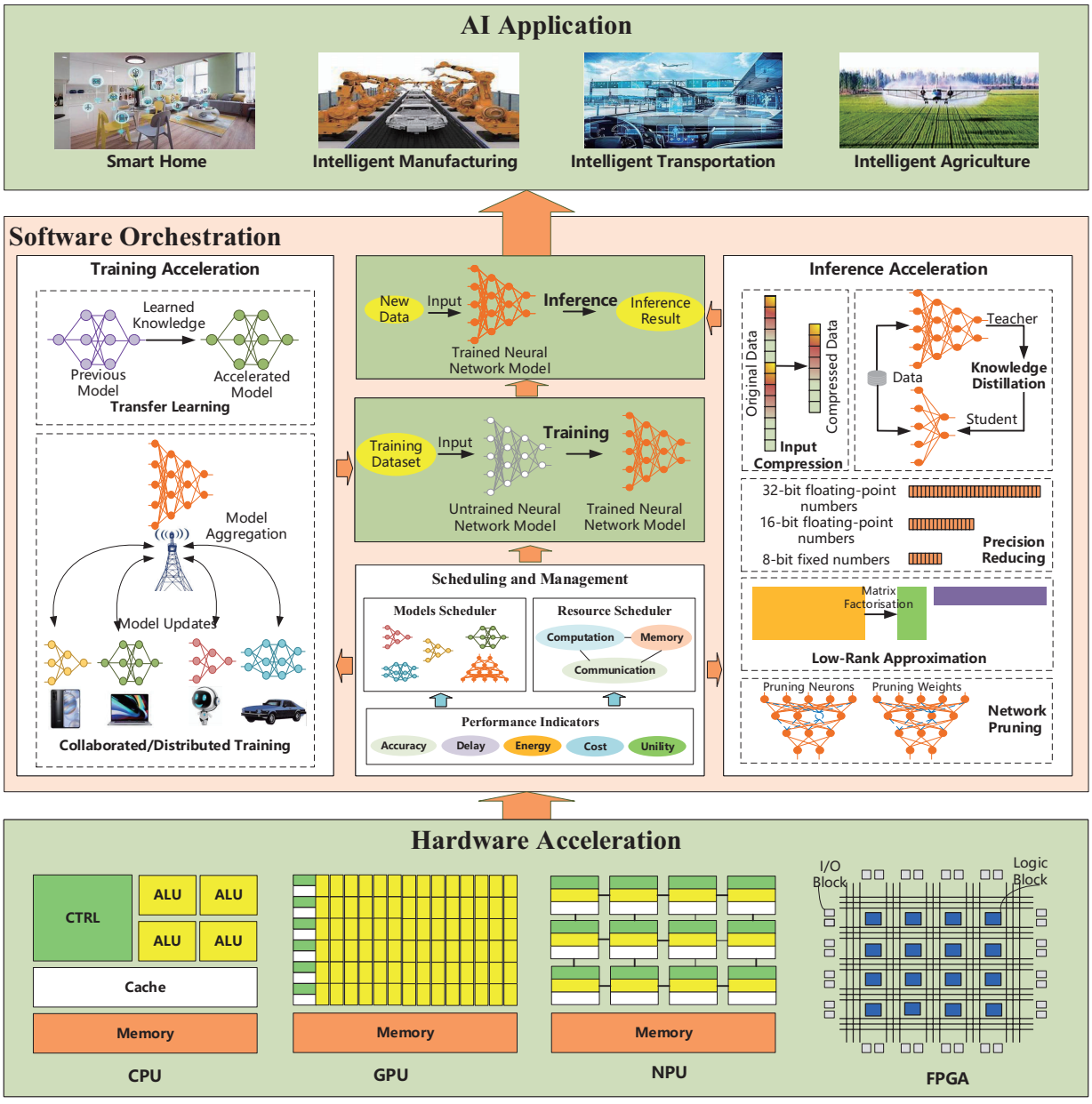


Figure 1. The edge AI computing framework with co-acceleration of software orchestration and hardware.

2) GPUs: The GPU is currently the most widely used for graphics processing for various kinds of games and the most flexible AI chip available in the AI chips market, which is much faster than general-purpose chips. GPUs with thousands of computational cores can achieve 10∼100x computing power compared to CPUs alone [6]. They are built with parallelism and very good at running AI algorithms with high parallelism demands.

3) FPGAs: Unlike CPUs and GPUs, FPGAs can be programmed with software based on their requirements or changed their functionality dynamically to satisfy the ever-changing demands. Reprogramming can avoid potential technology obsolescence of the ASIC-based approach. Generally, FPGAs have a shorter development cycle than ASIC-based AI chips and lower power requirements than GPUs. Therefore, FPGA with millions of parallel system logic cells is used primarily for inference, and it is suitable for implementing/optimizing AI algorithms. FPGA with enough flexibility and scalability can support a wide range of neural networks. However, FPGAs' high flexibility probably needs professional programming experience and special tools.

4) ASICs: ASIC-based AI chips are currently the most popular form of AI chips due to their unique optimization features for accelerating AI computing. These features [6] include a large number of calculations in parallel rather than sequentially, accelerating memory access by moving the computation inside or closer to memory (e.g., increasing the capacity of the on-chip memory and embedding a well-trained AI algorithm inside the memory), calculating numbers with lower precision to reduce the number of transistors while saving an enormous amount of memory and energy consumption, or designing chips with fully hardware-implemented deep neural networks in an energy-efficient way. Compared with GPUs and FPGAs, ASIC-based AI chips have a longer development cycle, less flexibility, smaller die size, higher efficiency, and lower power consumption. In Fig. 1, we take the architecture of Neural Processing Unit (NPU) as an example to visually illustrate the unique optimization features of ASIC-based AI chips. At present, some manufacturers have introduced different ASIC-based processors intended for various and computation-intensive tasks with high efficiency and high performance. For example, HiSilicon's 900 series chips introduce an additional NPU to accelerate the calculation of vectors and matrices, significantly improving the computational efficiency of AI. Google has developed a mobile AI chip (i.e., Tensor Processing Unit (TPU)) for its new smartphones named Google Pixel 6 to vastly improve the photo, video, text-to-speech and other features on mobile terminals. Horizon Robotics offers AI-based solutions for automotive applications powered by its Brain Processing Unit (BPU).

Actually, due to the diversity of requirements, it is challenging to design an all-purpose AI chip. Therefore, AI chips should vary in size, efficiency, speed, generality and accuracy, and they should

Table I

COMPARISON OF KEY AI CHIPS AND GENERAL-PURPOSE CHIPS.

Chip Types	CPU	GPU	FPGA	ASIC
Characteristics	Used for service control and management, suitable for processing control tasks with complex processed.	Used for graphics-intensive tasks processing, suitable for processing large-scale parallel computing tasks.	Programmable hardware, shorter development period than that of dedicated chips.	Dedicated to neural networks and customizable requirements based on AI applications.
Customization	General-purpose	Semi-customized	Semi-customized	Customized
Training Speed	Low	High	-	High
Inference Speed	Low	Medium	High	Very High
Maximum Accuracy	High	High	High	Low
Key Manufaturers	AMD, Intel, Nvidia	AMD, Nvidia	Intel, Xilinx	Google, Intel, Huawei

be industry-oriented and customized for various AI application scenarios. For example, autonomous vehicles require quite powerful AI chips to perform AI inference at network edge instead of in cloud for the nanoseconds level latency, while wearable devices' chips must process AI tasks under the strict constraints of power consumption. Customizing an AI chip for a specific industrial application requires lots of efforts in academia and industry, covering all hierarchies from algorithms, circuits to materials and manufacture, providing the feasibility of significantly improving performance and reducing service latency for AI applications. Besides, to break the performance bottlenecks of a single type of chip, coupling multiple AI chips, such as Nvidia's Grace AI chip coupled with ARM CPU and GPU, may gain more advantages in accelerating AI computing.

C. Software Orchestration

Due to weak computing power and limited battery life, mobile terminals are usually unable to afford computation-intensive and latency-sensitive AI tasks. Relying only on hardware acceleration to realize edge intelligence limits the computational flexibility, so we should also seek more flexible strategies from the perspective of software orchestration as follows.

1) Training acceleration: Training an AI model is quite time-consuming and computationally intensive, which is unaffordable for ordinary mobile terminals due to their limited memory, computing power, and energy. Therefore, it is necessary to take some measures to accelerate the training process, including Transfer Learning (TL) [7] and Federated Learning (FL) [8] shown in Fig. 1. In TF, learned knowledge on previous models can be transferred to new scenarios, thus significantly reducing the training costs. FL is a kind of distributed collaborative learning, and it requires mobile terminals to download the global model from the server and then upload the updated model to the server instead of uploading raw large-

scale training dataset to the server, thereby significantly reducing the communication costs and avoiding possible privacy leakage caused by uploading the raw training dataset.

2) Inference acceleration: Generally, the more AI model parameters (i.e., neurons and weights), the larger the AI model's size, and the higher the AI model's accuracy. However, large AI scale restricts the computing efficiency of AI models and further leads to dramatic computation costs such as high latency, memory consumption, and energy consumption. Therefore, it becomes essential to accelerate these models before deploying on resource-constrained mobile terminals while maintaining high model accuracy. Great efforts are trying to lighten AI computing load via input parameter compression and model optimizations. Effective pre-processing approaches can achieve input parameter compressions, such as narrowing searching space [9] and Region-of-Interest (RoI) encoding [10], which could significantly reduce the bandwidth consumption and data transmission latency when performing inference at the edge server. Since trained AI models are usually over-parameterized, researchers have made a lot of efforts on compression approaches to facilitate model optimizations [11] [12], such as network pruning by keeping only essential connections and deleting unimportant parameters like connections with fewer weights, low-rank approximation/matrix factorization by decomposing a matrix into the multiplication of multiple small-size matrices, knowledge distillation and parameter quantizing by eliminating redundant parameters, and reducing precision by using lower floating-point numbers. These model compression approaches can be applied to different AI models or can be composed to optimize a complex AI model to break through a single method's bottleneck on some performance improvements.

3) Model Scheduling and Resource Management: Given network resources, AI models, and hardware support, how to jointly schedule and manage limited network resources and available AI models to meet diversified performance metrics of AI applications, needs to be carefully considered.

Performance Metrics: Different AI application scenarios have different performance metric requirements such as accuracy, latency, energy consumption, reliability, cost, privacy, service revenue, flexibility, and scalability. For example, a training task usually requires higher computation precision, which directly influences the AI model's accuracy, while a security-like and latency-sensitive task (e.g., real-time video analysis) usually has higher requirements on inference accuracy and latency. When multiple performance metrics are involved in a decision, a new and urgent problem will arise: accurately identifying complex interactions and balancing multiple metrics to maximize the comprehensive performance when optimizing the AI model, determining the best offloading strategy and AI model, and optimizing resources allocation.

AI Model Scheduling: Due to different computing complexity and performance requirements, different AI models are designed for different application scenarios, such as feature extraction and function

approximation using fully connected neural network (FCNNs), image recognition using convolution neural networks (CNNs), machine translation and speech recognition using recurrent neural networks (RNNs). Even for the same AI algorithm, it will still have different performances because of different network parameters (e.g., the number of neurons). In this case, we must meet the challenges ahead and make greater efforts on AI model scheduling and decision. By comprehensively weighing different AI models' key metrics (e.g., accuracy and latency) and network conditions (e.g., hardware support, network traffic, and bandwidth), we can determine the best one from the available AI models library. Besides, combining advanced compression techniques (e.g., parameter quantizing and network pruning), optimal resource management and task scheduling strategies based on optimization/AI methods can potentially improve AI performance.

Resource Scheduling and Management: Since edge intelligence for ubiquitous AI seamlessly merges wireless communications and mobile edge computing, it faces resource allocation problems in different dimensions, including computation, memory, and communication resources. Compared with the cloud, resources of the network edge and mobile terminals are relatively fewer. To maximize resource utilization and optimize computing performance under the constraints of limited resources, tight coordination and high inter-operability of multi-dimensional resources are essential. Different AI applications have notable differences in resource requirements due to their respective performance objectives. Therefore, we should set different resource scheduling rules according to available resources and different quality of service (QoS) requirements of various AI tasks. Particularly, the resource scheduling criterion emphasizes accuracy, parallelism, and data volume for AI training, while the latency, energy efficiency and cost for AI inference.

III. A Case Study of Accelerating AI Inference

With increasingly diverse service and QoS requirements, it is critical to explore architectures and designs of AI computing systems to effectively support different service requirements. There are many limitations for running various AI tasks on mobile terminals, such as limited computing power, memory, and power. Therefore, it becomes more realistic to explore the diverse service and QoS requirements and dynamic network environment and then focus on the practical AI deployment in the edge computing architecture to meet various service requirements, such as accuracy and latency. As shown in Fig. 2, we set two different real-time AI tasks on two mobile terminals respectively as an example case to illustrate the edge AI inference architecture with co-acceleration of software orchestration and hardware. In particular, Task 1 on mobile terminal 1 is an image classification task that needs to be resized before it can be used

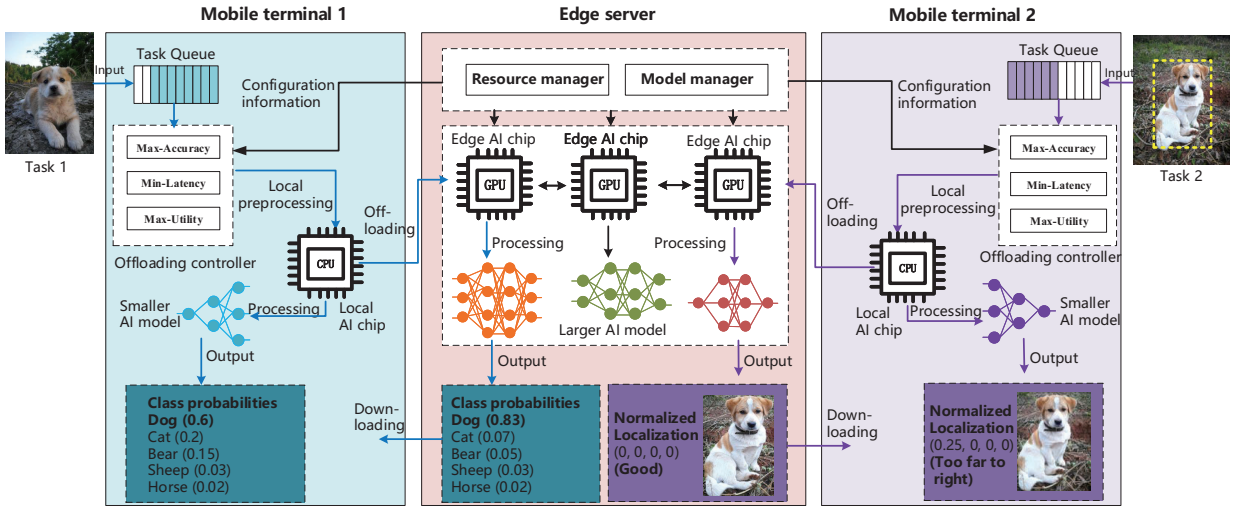


Figure 2. A case architecture of edge AI system.

as the input to the AI model. After AI inference, the confidence scores of multiple predefined classes can be obtained to identify what is in the image. The confidence score is in the range of 0∼1 and shows the probability of a detected object belonging to a predefined class. When the confidence score is greater than a predefined threshold, the inference result is right, otherwise it is wrong. Task 2 on mobile terminal 2 is an image object localization task and performs similar resizing processing to Task 1 before inputting the AI model. Unlike Task 1, the inference result for Task 2 shows where the bounding box accurately contains the object (i.e., the dog). The bounding box can be defined by a point, width, and height, (e.g., (x0, y0, width, height)). The edge bounding box (0, 0, 0, 0) accurately covers the dog, while the local bounding box (0.25, 0, 0, 0) may offset by 0.25 above the X-axis, which indicates that the bounding box is too far to the right.

Each mobile terminal is equipped with a hardware processor (i.e., CPU), and due to the limited computing power and battery life, it can only handle lightweight latency-tolerant AI tasks and cannot guarantee the service latency requirement of many latency-sensitive applications such as augmented reality and autonomous driving. Therefore, we consider that all mobile terminals transfer their latency-sensitive AI tasks to the more powerful MEC server and execute the deep learning algorithms there. Since the server is equipped with multiple parallel hardware accelerators (i.e., multiple GPU), and has more powerful computing power than mobile terminals, it can simultaneously process multiple large-scale AI tasks. Hardware acceleration methods, such as designing ASIC chips integrated with AI algorithm and reducing floating-point numbers, essentially accelerate AI processing by increasing available computing

power, which usually directly affect the processing latency [13].

When a new AI task arrives, it will first be stored in the task queue, waiting to be processed on the MEC server. Each mobile terminal starts task offloading only after the local preprocessing has completed, and starts edge computing only after the task offloading has finished. There are multiple candidate AI models with different sizes on the MEC server to support a wide variety of QoS requirements (e.g., latency, accuracy, and energy consumption), where a smaller AI model can reduce the processing latency and energy consumption at the cost of accuracy, and a larger AI model can increase the accuracy at the cost of longer processing latency and higher energy consumption [14] [15]. Based on the local controller (e.g., Max-Accuracy, Min-Latency, or Max-Utility) and current network conditions (e.g., network traffic or pre-configurations/configurations like bandwidth allocation from the resource manager and model manager), a mobile terminal can choose the appropriate one from available AI models.

To highlight the advantages of the proposed framework, we give some performance comparisons through simulation. In our simulation, we assume that there are 6 mobile terminals and one BS with an MEC server distributed within a squared area of 1000 $\times$1000 m^2. The coordinate of the BS is (500m, 500m), and mobile terminals are randomly distributed in this squared zone and associated to the BS. The total wireless channel bandwidth is set as 20 MHz, and we consider that all mobile terminals offload their AI tasks via orthogonal frequency division multiplexing channels (OFDMA) without interference to the MEC server. We consider that the uplink transmission channel is the Rayleigh channel and the noise power density is -174 dBm/Hz. The transmission power and energy budget of each mobile terminal are set as 0.3 W and 0.05 J, respectively. The computation capability of each mobile terminal and the MEC server are set to 1$\sim$2 GHz and 14 GHz, respectively. Similar to those in [15], the required CPU cycles per bit is set as 1000 CPU cycles/bit, and the effective switched capacitance is set as 10^{-28}. There are three AI models deployed on the edge server. Corresponding to each AI model, the original input sizes are set to 100×100, 300×300 and 600×600 pixels, respectively, and the data size per pixel is set as 24 bits/pixel. Assume that the processing latency of each task is required to be less than or equal to 0.5 s.

AI task is latency-sensitive and energy-consuming, generally requiring high learning accuracy. We use the existing latency and energy consumption model in [15]. According to the existing works [14] [15], the learning accuracy generally increases with the input model size under a fixed AI architecture, at the cost of longer processing latency and higher energy consumption. Therefore, a mobile terminal can obtain different accuracy values by selecting different models with different sizes. To quantify the accuracy level of the whole system, we define the accuracy performance indicator of the system as the

ratio of the sum of the desired model decisions to the sum of the model selection decisions. In our simulations, we focus on minimizing the total service latency of all latency-sensitive AI tasks with the learning accuracy, task processing latency and energy consumption constraints, by jointly optimizing the AI model decision, computation resource and communication bandwidth allocation. We evaluate the performance of the proposed scheme by comparing it with the following schemes:

• Greedy Latency scheme denotes that the AI model and network resource allocation can be decided by minimizing the latency regardless of the accuracy constraints.

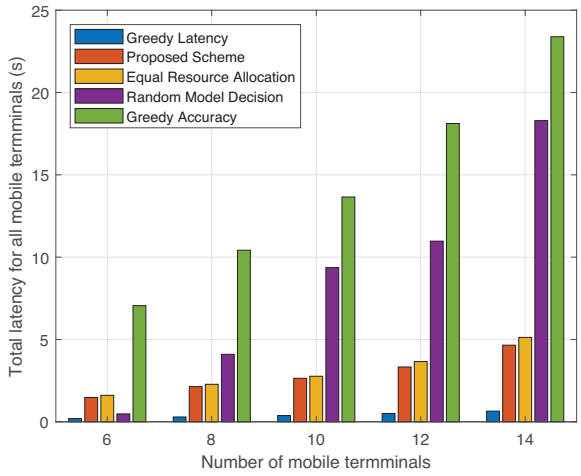
• Greedy Accuracy scheme denotes that the AI model and resource allocations can be decided by maximizing the accuracy while the energy consumption and latency constraints are ignored.

• Random AI Model Decision scheme denotes that the AI model can be decided randomly for each mobile terminal while network resources are allocated optimally.

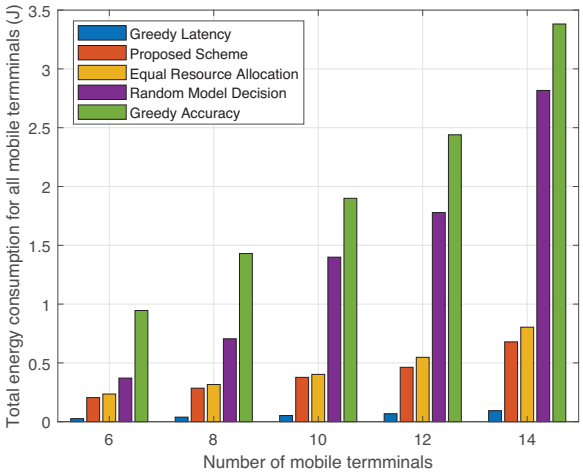
• Equal Resource Allocation scheme denotes that network resources allocated to each mobile terminal are equal while the AI model is selected optimally.

Figure 3 compares the total service latency, accuracy, and energy consumption between the proposed scheme and other schemes. It can be seen that under different numbers of mobile terminals, our proposed scheme always shows better latency and energy consumption performance than Greedy Accuracy, Random Model Decision, and Equal Resource Allocation schemes. As the number of mobile terminals increases, the performance gain of our scheme becomes more prominent. Our scheme can achieve lower latency and energy consumption by jointly optimizing the model decision and computing and communication resource allocation while ensuring learning accuracy requirement, latency and energy consumption limitations. From Fig. 3, Greedy Latency scheme has the lowest latency, energy consumption and accuracy, which is due to the fact that Greedy Latency scheme always chooses the smallest AI models without considering significant accuracy loss. By contrast, Greedy Accuracy scheme always chooses the largest AI models for maximum accuracy, thus resulting in the highest processing latency and energy consumption.

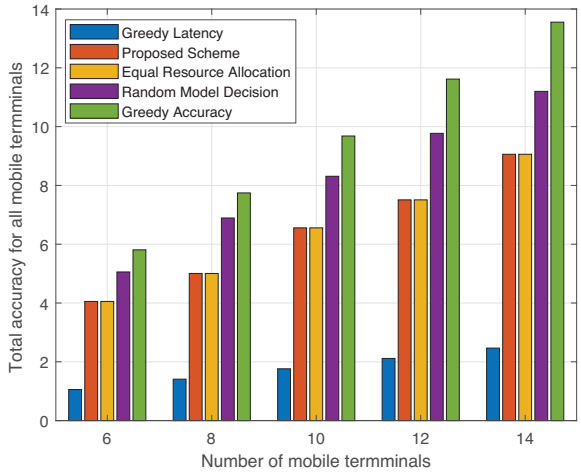
Available computing power could be increased through effective hardware acceleration methods, such as designing ASIC chips integrated with AI algorithm and reducing floating-point numbers [13]. Fig. 4 depicts the total service latency versus available computing power. Intuitively, the service latency reduces as the available computing power increases. Besides, Fig. 4 shows the total service latency under different model input sizes and desirable accuracy requirements. $\boldsymbol{\phi_1}$ and $\boldsymbol{\phi_2} = \{90 \times 90, 280 \times 280, 570 \times 570\}$ (pixels) are the original model size set and the optimized model size set after re-designing or optimizing, such as parameter pruning. It can be seen that the service latency reduces as the model input size reduces. $\boldsymbol{\chi} = \{\chi_1 = 0.67, \chi_2 = 0.75\}$ denotes the desirable accuracy indicator of the system. The larger the $\boldsymbol{\chi}$,



(a)



(b)



(c)

Figure 3. Comparison of total service latency, total accuracy and total energy consumption of different schemes under different numbers of mobile terminals: a) comparison of total service latency of different schemes; b) comparison of total accuracy of different schemes; c) comparison of total energy consumption of different schemes.

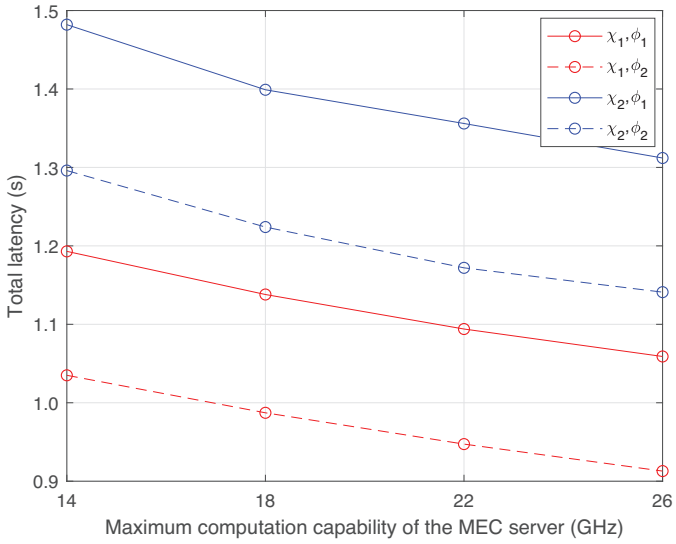


Figure 4. Total service latency versus different hardware computing power and learning accuracy requirements.

the higher the learning accuracy level of the system, which requires mobile terminals to select larger-size AI models under the constraints of energy consumption and processing latency to guarantee the learning accuracy requirements, but it will consume much more computing resources to reduce the service latency.

It can be also seen that, using a series of acceleration solutions including customizing an AI chip integrated with AI algorithm, optimizing AI model decisions and resource allocation, can further accelerate AI computing.

IV. Conclusions and Future Work

In this article, we have proposed a low-latency edge AI computing networking framework to promote the development of edge intelligence, aided by co-acceleration of software orchestration and hardware, including using advanced and customized AI chips and model optimization methods. We also have shown the potential benefits of the proposed framework via simulation.

Although recent researches on AI accelerations have made remarkable progresses in many fields, such as model optimization methods and AI chip designs, some issues are still worth further exploring to realize edge intelligence. For example, combining multiple AI model optimization methods to break through a single optimization method's performance bottleneck can further boost the overall system performance. However, this may require a thorough understanding of the complex interaction between different optimization methods. Besides, edge intelligence relies on the direct interaction between mobile terminals and MEC servers through wireless communications, which poses a great challenge to the maintenance and

management of edge intelligence involving complex wireless communication environments and limited network resources.

ACKNOWLEDGEMENTS

The work was supported in part by NSFC under Grant 62071393, NSFC High-Speed Rail Joint Foundation under Grant U1834210, Sichuan Provincial Applied Basic Research Project under Grant 2020YJ0218, and Fundamental Research Funds for the Central Universities under Grant 2682021CF019.

REFERENCES

[1] X. Wang *et al.*, "In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning," *IEEE Network*, vol. 33, no. 5, Sept.-Oct. 2019, pp. 156-165.

[2] S. Deng *et al.*, "Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence," *IEEE Internet of Things J.*, vol. 7, no. 8, Aug. 2020, pp. 7457-7469.

[3] S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," https://arxiv.org/abs/1510.00149, Oct. 2015.

[4] E. Li *et al.*, "Edge AI: On-Demand Accelerating Deep Neural Network Inference via Edge Computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, Jan. 2020, pp. 447-457.

[5] X. Wang *et al.*, "Net-in-AI: A Computing-Power Networking Framework with Adaptability, Flexibility, and Profitability for Ubiquitous AI," *IEEE Network*, vol. 35, no. 1, Dec. 2020, pp. 280-288.

[6] S. Khan and A. Mann, "AI Chips: What They Are and Why They Matter," *Georgetown Center for Security and Emerging Technology*, https://cset.georgetown.edu/research/ai-chips-what-they-are-and-why-they-matter/, Apr. 2020.

[7] O. Valery, P. Liu, and J. Wu, "CPU/GPU Collaboration Techniques for Transfer Learning on Mobile Devices," *Proc. IEEE ICPADS*, Shenzhen, China, 2017.

[8] M. Chen *et al.*, "A Joint Learning and Communications Framework for Federated Learning Over Wireless Networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, Jan. 2021, pp. 269-283.

[9] S. Y. Nikouei *et al.*, "Real-Time Human Detection as an Edge Service Enabled by a Lightweight CNN," *Proc. IEEE EDGE*, San Francisco, CA, USA, 2018.

[10] L. Liu, H. Li, and M. Gruteser, "Edge Assisted Real-Time Object Detection for Mobile Augmented Reality," *Proc. MobiCom*, Los Cabos, Mexico, 2019.

[11] R. Xie *et al.*, "Energy Efficiency Enhancement for CNN-based Deep Mobile Sensing," *IEEE Wireless Commun.*, vol. 26, no. 3, June 2019, pp. 161-167.

[12] T. Choudhary *et al.*, "A Comprehensive Survey on Model Compression and Acceleration," *Artif. Intell. Rev.*, vol. 53, no. 7, Oct. 2020, pp. 5113-5155.

[13] T. Tan and G. Cao, "FastVA: Deep Learning Video Analytics Through Edge Processing and NPU in Mobile," *Proc. IEEE INFOCOM*, Toronto, Canada, July 2020.

[14] C. Wang *et al.*, "Joint Configuration Adaptation and Bandwidth Allocation for Edge-based Real-time Video Analytics," *Proc. IEEE INFOCOM*, Toronto, Canada, July 2020.

[15] Y. He *et al.*, "Optimizing the Learning Performance in Mobile Augmented Reality Systems With CNN," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, Aug. 2020, pp. 5333-5344.

Cailian Deng (dengcailian@my.swjtu.edu.cn) received the B.E. degree in communication engineering from Southwest Jiaotong University, Emei, China, in 2017. She is currently working toward the Ph.D. degree with the Key Laboratory of Information Coding and Transmission, School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China. Her research interests include radio resource management, edge intelligence, mobile edge computing, integrated communication and sensing.

Xuming Fang [M'00, SM'16] (xmfang@swjtu.edu.cn) is a professor at Southwest Jiaotong University, China. He held visiting positions with the Technical University Berlin, Berlin, Germany, and with the University of Texas at Dallas, Richardson, TX, USA. He has published over 200 high-quality research papers in journals and conference publications. His research interests include wireless broadband access control, radio resource management, mobile edge computing, integrated communication and sensing, and broadband wireless access for high-speed railways.

Xianbing Wang [S'98-M'99-SM'06-F'17] (xianbin.wang@uwo.ca) received the Ph.D. degree in electrical and computer engineering from National University of Singapore, Singapore, in 2001. He is currently a Professor and the Tier-I Canada Research Chair at Western University, London, ON, Canada. Prior to joining Western, he was with Communications Research Centre Canada as a Research Scientist/Senior Research Scientist between July 2002 and December 2007. From January 2001 to July 2002, he was a System Designer at STMicroelectronics. His research interests include 5G technologies, Internet-of-Things, mobile edge computing, machine learning, and locating technologies. He has more than 350 peer-reviewed journal and conference papers.

Kevin Law (k.law@huawei.com) is Principal Architect in Optics Production Line of Huawei, China. His research interests include intelligent IC design, intelligent computing, optical communications, mobile communications and applied mathematics.

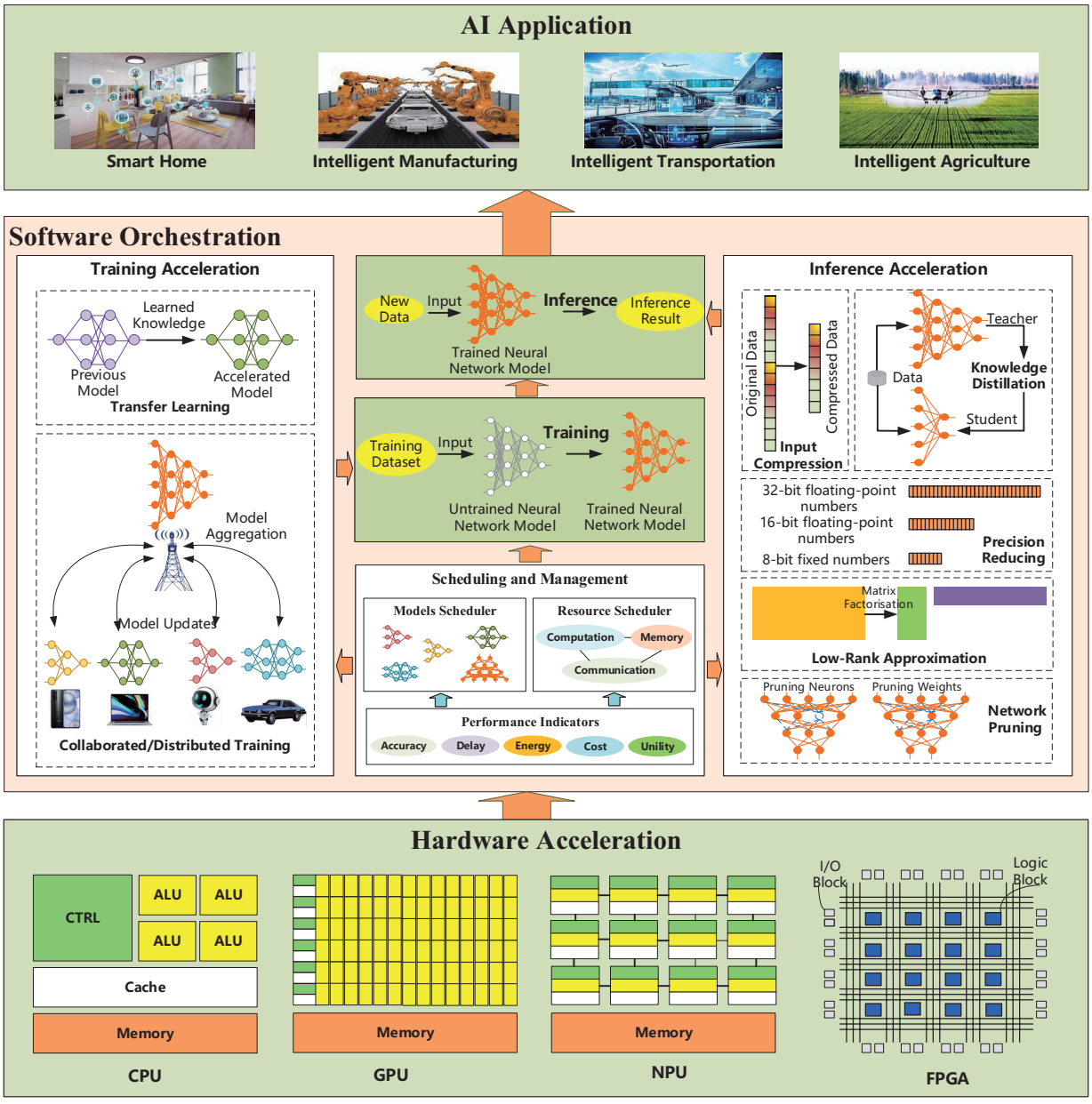


Figure 1. The edge AI computing framework with co-acceleration of software orchestration and hardware.

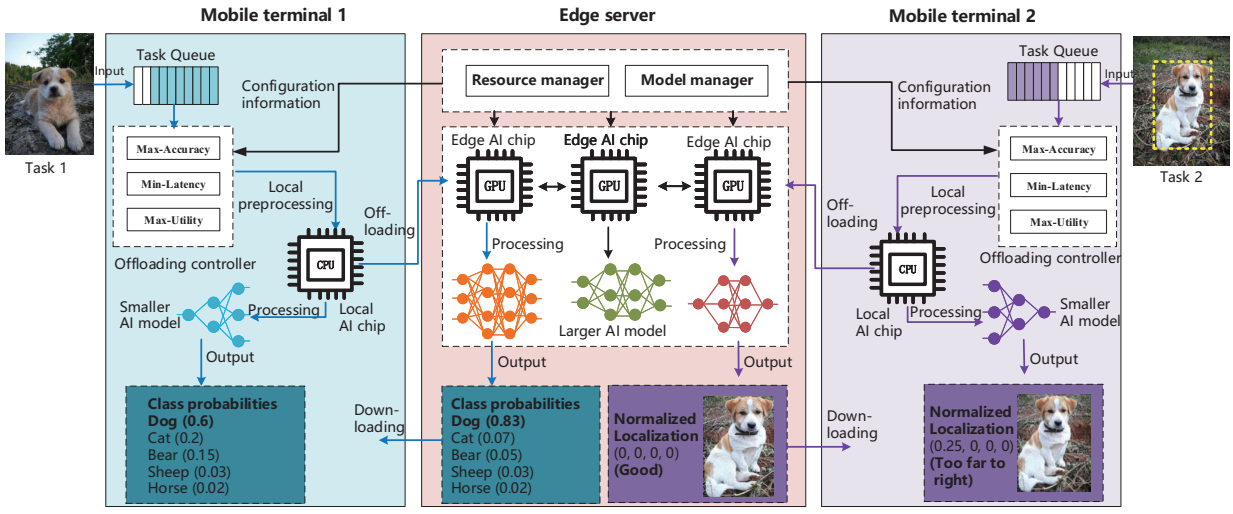
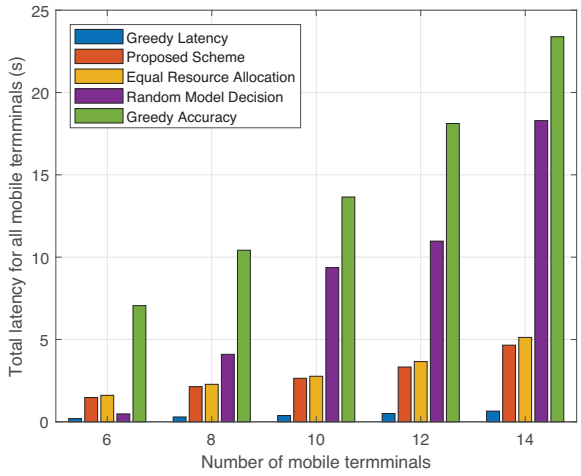
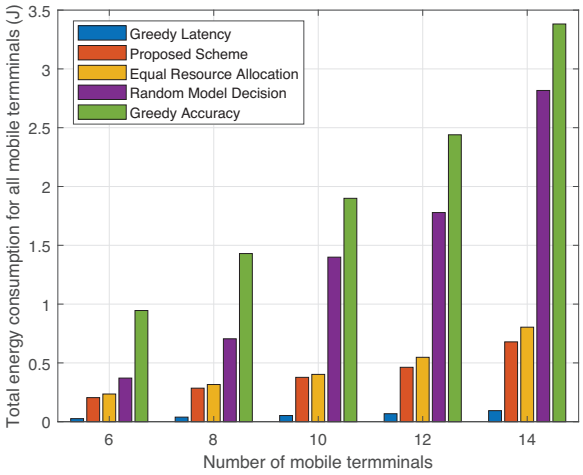


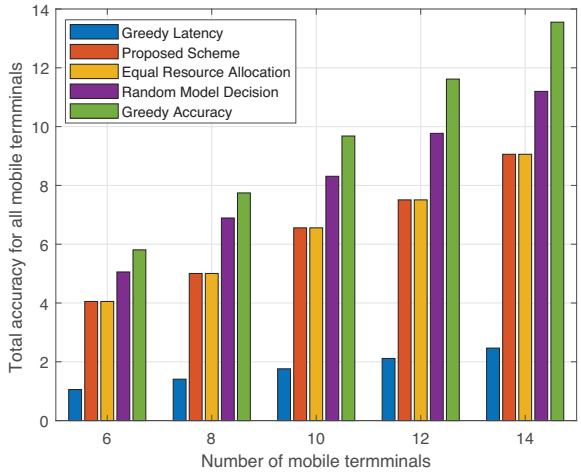
Figure 2. A case architecture of edge AI system.



(a)



(b)



(c)

Figure 3. Comparison of total service latency, total accuracy and total energy consumption of different schemes under different numbers of mobile terminals: a) comparison of total service latency of different schemes; b) comparison of total accuracy of different schemes; c) comparison of total energy consumption of different schemes.

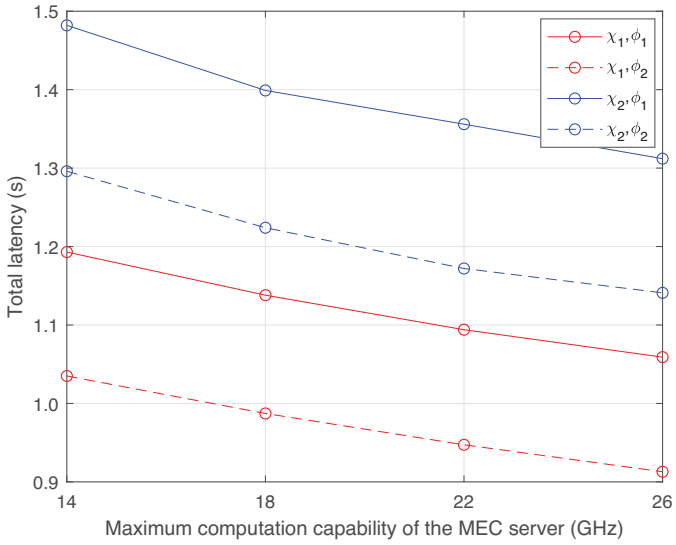


Figure 4. Total service latency versus different hardware computing power and learning accuracy requirements.