Empowering Edge Intelligence: A Comprehensive Survey on On-Device AI Models

XUBIN WANG, Hong Kong Baptist University, Beijing Normal Hong Kong Baptist University and Beijing Normal University, China
ZHIQING TANG, Beijing Normal University, China
JIANXIONG GUO, Beijing Normal University and Beijing Normal Hong Kong Baptist University, China
TIANHUI MENG, Beijing Normal Hong Kong Baptist University, China
CHENHAO WANG, Beijing Normal University and Beijing Normal Hong Kong Baptist University, China
TIAN WANG, Beijing Normal University, China
WEIJIA JIA, Beijing Normal University and Beijing Normal Hong Kong Baptist University (Corresponding author), China

The rapid advancement of artificial intelligence (AI) technologies has led to an increasing deployment of AI models on edge and terminal devices, driven by the proliferation of the Internet of Things (IoT) and the need for real-time data processing. This survey comprehensively explores the current state, technical challenges, and future trends of on-device AI models. We define on-device AI models as those designed to perform local data processing and inference, emphasizing their characteristics such as real-time performance, resource constraints, and enhanced data privacy. The survey is structured around key themes, including the fundamental concepts of AI models, application scenarios across various domains, and the technical challenges faced in edge environments. We also discuss optimization and implementation strategies, such as data preprocessing, model compression, and hardware acceleration, which are essential for effective deployment. Furthermore, we examine the impact of emerging technologies, including edge computing and foundation models, on the evolution of on-device AI models. By providing a structured overview of the challenges, solutions, and future directions, this survey aims to facilitate further research and application of on-device AI, ultimately contributing to the advancement of intelligent systems in everyday life.

CCS Concepts: • **General and reference → Surveys and overviews**; • **Computing methodologies → Artificial intelligence**; **Machine learning**.

Additional Key Words and Phrases: On-Device AI, Edge Intelligence, Real-time Processing, Model Optimization, Data Privacy, Survey.

Authors' Contact Information: Xubin Wang, wangxubin@ieee.org, Hong Kong Baptist University, Beijing Normal Hong Kong Baptist University and Beijing Normal University, China; Zhiqing Tang, zhiqingtang@bnu.edu.cn, Beijing Normal University, Zhuhai, Guangdong, China; Jianxiong Guo, jianxiongguo@bnu.edu.cn, Beijing Normal University and Beijing Normal Hong Kong Baptist University, Zhuhai, Guangdong, China; Tianhui Meng, tmeng@bnu.edu.cn, Beijing Normal Hong Kong Baptist University, Zhuhai, Guangdong, China; Chenhao Wang, chenhwang@bnu.edu.cn, Beijing Normal University and Beijing Normal Hong Kong Baptist University, Zhuhai, Guangdong, China; Tian Wang, tianwang@bnu.edu.cn, Beijing Normal University, Zhuhai, Guangdong, China; Weijia Jia, jiawj@bnu.edu.cn, Beijing Normal University and Beijing Normal Hong Kong Baptist University (Corresponding author), Zhuhai, Guangdong, China.

ACM 1557-7341/2025/3-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Xubin Wang, Zhiqing Tang, Jianxiong Guo, Tianhui Meng, Chenhao Wang, Tian Wang, and Weijia Jia. 2025. Empowering Edge Intelligence: A Comprehensive Survey on On-Device AI Models. *ACM Comput. Surv.* 1, 1 (March 2025), 42 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

In the past decade, the rapid development of AI technology has led to the widespread application of AI models across various fields [10, 26]. From AlphaGo to ChatGPT, these breakthrough advancements demonstrate the immense potential of AI in different domains [150]. However, despite significant achievements, deploying AI applications in real-world settings remains challenging due to factors such as high computational demands, scalability, and privacy concerns [249, 277]. In this context, large language models like GPT-3, which boasts 175 billion parameters and requires approximately 800GB of storage [18], demonstrate remarkable capabilities. Nevertheless, their substantial size poses limitations for deployment on devices.

Traditionally, AI models have relied on powerful cloud computing resources for training and inference [49]. However, with the proliferation of the IoT, edge computing, and mobile devices, an increasing number of AI models are being deployed on-device [42, 193]. This shift not only enhances the real-time processing and efficiency of data handling but also reduces reliance on network bandwidth and strengthens data privacy protection [60]. Specifically, Gartner projects that by 2025, approximately 75% of all enterprise-generated data will be produced outside traditional data centers [60]. Transmitting and processing this data in centralized cloud systems introduces significant system and latency overhead, along with substantial bandwidth requirements [43]. This also underscores the importance of deploying AI models on-device.

Edge intelligence enhances the concept of localized data processing by deploying AI algorithms directly on edge devices, thereby reducing reliance on cloud infrastructure [284]. This approach not only facilitates faster data processing but also addresses important privacy and security concerns, as sensitive data remains within the local environment [57, 122, 203], with on-device AI models finding application in various scenarios, such as smartphones, smart home systems, autonomous vehicles, and medical devices [43]. However, the effective implementation of AI models on edge devices poses significant challenges. The reliance of these models on large parameter counts and powerful processing capabilities necessitates the development of innovative strategies for model compression, optimization, and adaptation to specific operational environments [42]. Addressing these challenges is crucial for maximizing the potential of edge intelligence in real-world applications.

While implementing efficient on-device AI models holds promise, it necessitates performance trade-offs, as optimizing models for constrained environments often involves sacrificing model accuracy or scalability to maintain functionality [20]. Thus, in light of these constraints, there is a growing imperative to design AI models that are both computationally efficient and adaptable to edge environments [27, 158]. These advances would facilitate the broader application of AI in fields such as Industry 4.0, where real-time, automated data processing is critical for monitoring, risk detection, and optimizing factory operations [11]. The successful implementation of such application has driven the proliferation and intelligence of smart devices, transforming people's lifestyles and work patterns [43]. Therefore, in-depth research into the characteristics, applications, and challenges of on-device AI models is of significant importance for advancing the development and application of AI technology.

1.1 Definition of On-Device AI Models

On-device AI models refer to AI models that are designed, trained, and deployed on edge or terminal devices. These models can perform data processing and inference locally without the need to transmit data to the cloud for processing [44, 248]. On-device AI models typically possess the following characteristics:

- **Real-time Performance:** They can quickly respond to user requests, making them suitable for applications that require immediate feedback [11].
- **Resource Constraints:** They are limited in computational power, storage, and energy consumption, necessitating optimization to fit the hardware environment of the device [20].
- **Data Privacy:** By processing data locally, they reduce the risks associated with data transmission, thereby enhancing user privacy protection [284].

1.2 Research Questions and Structure Overview of the Survey

This review aims to comprehensively explore the current state, technical challenges, and future development trends of on-device AI models. Specifically, our focus is to provide an academic response to the following research questions (RQs):

- RQ1: What are the applications of on-device AI models in daily life?
- RQ2: What are the main technical challenges for deploying on-device AI models?
- RQ3: What are the most effective optimization and implementation methods for enhancing the performance of on-device AI models
- RQ4: What are the future trends of on-device AI models?

Through a review and analysis of relevant literature, this paper provides researchers and engineers with a clear perspective to help them understand the key issues and solutions related to on-device AI models. The structure of the review is as follows: *Section 2* introduces the fundamental concepts of on-device AI models and explains how they work. *Section 3* explores the application scenarios of on-device AI models, covering areas such as smartphones, IoT devices, and edge computing (RQ1). *Section 4* analyzes the technical challenges faced by on-device AI models, such as computational resource limitations, energy management, and data privacy issues (RQ2). *Section 5* discusses optimization and implementation methods for on-device AI models, including data optimization, model compression, and hardware acceleration techniques (RQ3). *Section 6* looks ahead to future development trends, exploring the impact of emerging technologies on on-device AI models (RQ4). *Section 7* summarizes the main findings of the review and provide suggestions for future research. The diagram in Figure 1 shows the overall framework and methodology employed in this survey.

1.3 Contributions of This Survey

This survey makes several key contributions to the field of on-device AI models:

(1) **Comprehensive Overview**: It provides a thorough examination of the current landscape of on-device AI models, synthesizing existing research and identifying gaps in the literature.
(2) **Identification of Challenges**: The survey highlights the critical technical challenges faced by on-device AI models, including resource constraints, energy efficiency, and privacy concerns, thereby guiding future research efforts.
(3) **Optimization Strategies**: It discusses various optimization techniques and implementation methods that can enhance the performance of on-device AI models, offering practical insights for researchers and practitioners.
(4) **Future Directions**: The survey outlines potential future research directions and emerging technologies that could influence the development of on-device AI models, encouraging innovation in this rapidly evolving field.
(5) **Practical Implications**: By addressing the real-world applications and implications of on-device AI models, this survey serves as a valuable resource for industry professionals looking to implement AI solutions in edge environments.

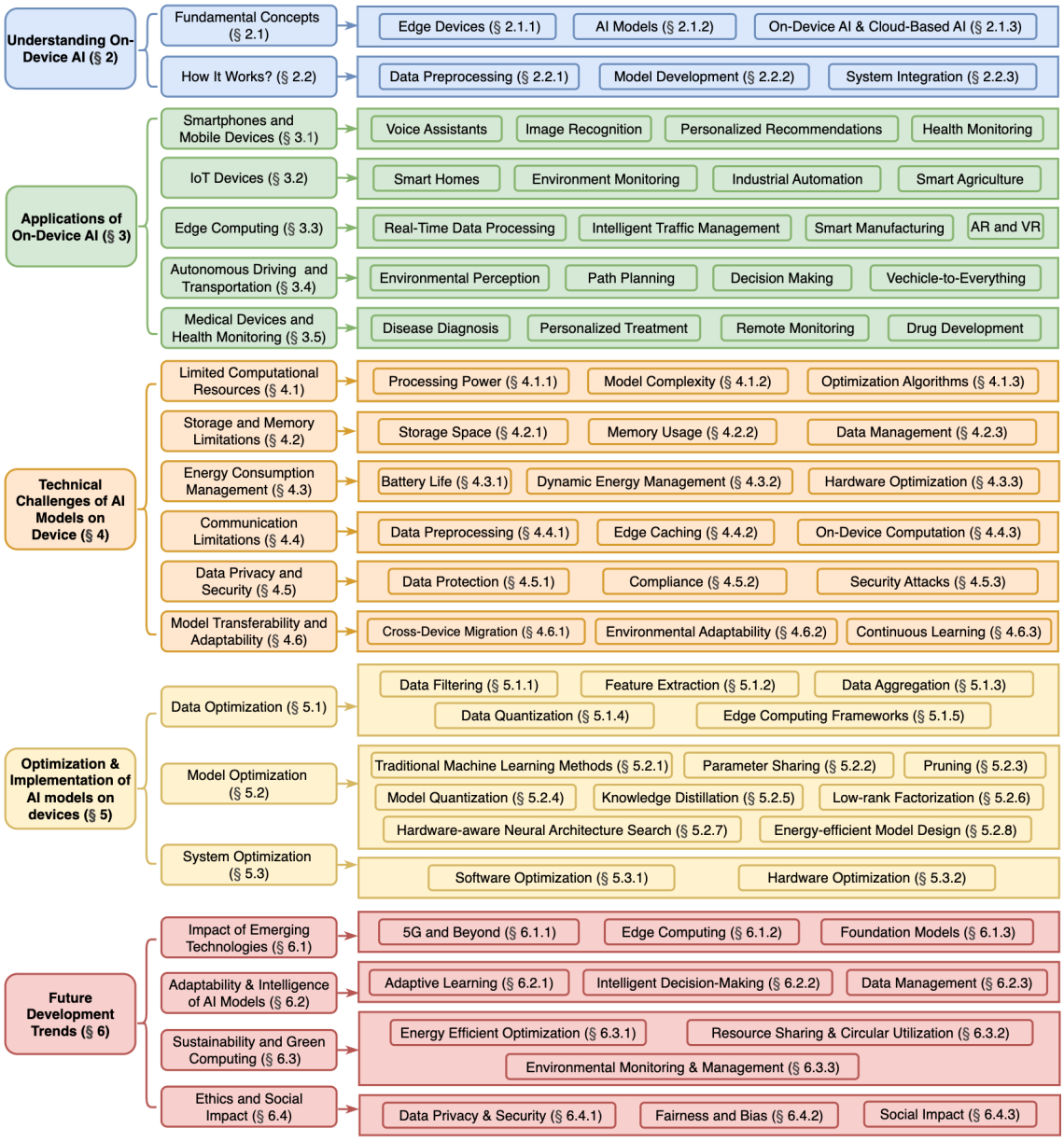

Fig. 1. Structure of this survey.

1.4 Related Surveys and Their Scope

Previous research has made significant contributions across diverse facets of on-device AI. Surveys by Shi *et al.* [188] and Dai *et al.* [40] have focused on efficient communication and computation offloading in edge systems, which are crucial for optimizing the performance of on-device AI models. Other studies, such as those by Zhang *et*

al. [266], have explored the application of mobile edge AI in vehicular networks, while Park *et al.* [169] provided an overview of wireless network intelligence that supports on-device AI functionalities. More recent investigations, including those by Deng *et al.* [43], have emphasized the dual roles of AI on edge devices and the support of edge functionalities, primarily focusing on frameworks and infrastructural requirements for deploying models closer to data sources (Xu *et al.* [247]; Murshed *et al.* [158]). Additionally, Xu et al. [248] introduced a survey about on-device language models, while Dhar et al. [44] provided a comprehensive overview of on-device machine learning (ML) from an algorithms and learning theory perspective. However, while these surveys lay essential groundwork, few offer an integrated perspective on deploying efficient on-device AI models specifically tailored to the constraints of edge environments. This gap underscores the necessity for a comprehensive review that not only summarizes advancements in on-device AI but also delves into the triad of optimization strategies for data, model, and system design (Chen *et al.* [27]; Zhou *et al.* [284]; Cai *et al.* [20]).

By addressing these challenges and exploring the potential of on-device AI models, this survey aims to contribute to the ongoing discourse in the field and facilitate the development of innovative solutions that can leverage the advantages of edge computing and IoT technologies.

2 Understanding On-Device AI Models

2.1 Fundamental Concepts of On-Device AI Models

2.1.1 Edge Devices. Edge devices encompass a wide range of hardware, from high-performance edge servers capable of executing complex computational tasks to resource-constrained IoT sensors designed for specific applications [190]. This category includes diverse devices such as smartphones, drones, autonomous vehicles, industrial robots, and smart home technologies, all of which are equipped to run AI models locally, facilitating real-time data processing [43]. The concept of edge computing, which emphasizes bringing services closer to the user, has its roots in the idea of cloudlets, as discussed by Satyanarayanan et al. [185]. Prominent hardware manufacturers, including NVIDIA and Intel, provide substantial support for the deployment of ML models on these edge devices, thereby enhancing their functionality for applications that demand low latency and high efficiency. For instance, NVIDIA's Jetson platform is widely recognized for its exceptional processing capabilities and is extensively utilized in edge AI applications [153]. Similarly, Intel's technologies enable seamless integration with IoT systems, promoting efficient data handling and analysis at the network's edge [31]. Table 1 presents a selection of edge devices along with their key features and typical use cases, illustrating the diversity and applicability of these technologies in various domains [50].

Table 1. List of Edge Devices and Their Features [50]

Device	Key Features	Use Cases
NVIDIA Jetson Xavier NX	6-core ARM CPU, 384-core GPU, 21 TOPS AI performance	Robotics, computer vision
Google Coral Dev Board	Edge TPU, 4 TOPS, low power consumption (2-4W)	Image recognition, object detection
Raspberry Pi 4	Quad-core ARM CPU, up to 8GB RAM, dual 4K HDMI output	IoT applications, home automation
AWS DeepLens	Intel Atom processor, integrated HD camera	Real-time computer vision
Intel NUC	Compact form factor, supports up to i7 processors	Digital signage, industrial automation
Microsoft Azure Stack Edge	Hybrid solution with GPU options	AI inferencing, video analytics
HPE Edgeline EL300	Rugged design, Intel Xeon/Core processors	Industrial applications
Lenovo ThinkEdge SE50	Intel Core i5/i7, rugged design	Smart cities, retail applications
Dell EMC PowerEdge XE2420	Dual Xeon processors, ruggedized chassis	Edge computing in harsh environments
Advantech MIC-770	Modular design, high-performance computing capabilities	Industrial edge applications

2.1.2 AI Models. At the core of on-device AI are AI models, which comprise algorithms specifically designed to interpret data and make decisions based on the information processed at the edge [261]. These models can

vary significantly in complexity, ranging from simple rule-based systems to sophisticated ML algorithms. By deploying AI models on edge devices, organizations can facilitate intelligent automation, predictive maintenance, and personalized user experiences, all while ensuring the maintenance of data privacy and security [44, 248]. The landscape of AI models has evolved considerably in recent years, particularly with the advent of foundation model technologies. This has resulted in the development of increasingly substantial models, such as Gemma 2B [215] and Llama 3.2 1B, which are specifically designed for deployment on edge devices [248]. AI models can be categorized into several types, each with distinct characteristics and applications. Table 2 summarizes these categories, providing a brief description and examples for each type:

Table 2. Types of AI Models

Type	Description	Examples
ML	Data-driven learning and prediction	Supervised, Unsupervised, Semi-supervised
Deep Learning	Multi-layer neural networks for pattern recognition	CNNs, RNNs
Reinforcement Learning	Trial-and-error learning through environment interaction	Game AI, Robotics
Transfer Learning	Applying knowledge from one domain to another	Fine-tuning pre-trained models

2.1.3 Comparison of On-Device AI Models and Cloud-Based AI Models. The comparison between on-device AI models and cloud-based AI models highlights several critical aspects that influence their deployment and effectiveness (see Table 3). On-device AI models are constrained by the computational resources available on the device, necessitating optimization to function efficiently; however, they offer lower latency, making them suitable for real-time applications [261]. In contrast, cloud-based AI models leverage powerful cloud infrastructure, enabling the support of complex models but often resulting in higher latency, which can be detrimental in time-sensitive scenarios [248]. Data privacy is another significant consideration, as on-device models enhance privacy by processing data locally, thereby reducing the risks of data breaches, while cloud-based models face higher security risks due to the transmission of data to external servers [215]. Scalability also differs markedly; on-device models have limited scalability due to hardware constraints, whereas cloud-based models can dynamically adjust resources to accommodate varying demands [190]. Finally, maintenance presents a contrasting challenge: on-device models require complex updates and maintenance, particularly in large deployments, while cloud-based models benefit from centralized management, simplifying the update and maintenance processes [43].

Table 3. Comparison of On-Device AI Models and Cloud-Based AI Models

Aspect	On-Device AI Models	Cloud-Based AI Models
Computational Resources	Limited by device capabilities; requires optimization	Utilizes powerful cloud resources; supports complex models
Latency	Lower latency; suitable for real-time applications	Higher latency; not ideal for time-sensitive scenarios
Data Privacy	Enhanced privacy; local data processing reduces breach risks	Higher security risks; data transmitted to the cloud
Scalability	Limited scalability; constrained by hardware capabilities	Good scalability; resources can be adjusted dynamically
Maintenance	Complex updates and maintenance in large deployments	Centralized management simplifies updates and maintenance

2.2 How On-Device AI Works

The process of implementing on-device AI involves a comprehensive pipeline that encompasses data processing, model development, and system integration, as illustrated in Figure 2. This figure provides an overview of the key components and workflow involved in deploying AI models at the edge device, highlighting the interplay between data, model, and system optimization.

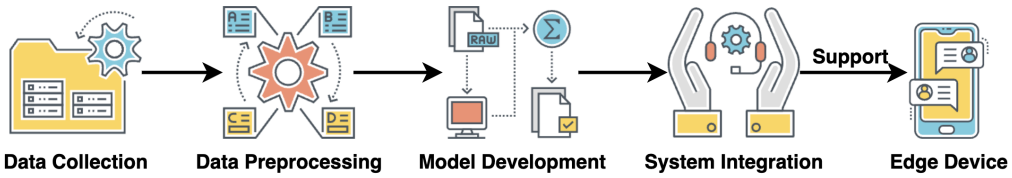

Fig. 2. An overview of how on-device AI works. The figure illustrates a general pipeline encompassing three critical aspects: data, model, and system. It is important to note that not all steps are necessary in practical applications.

2.2.1 Data Preprocessing. The first step in the on-device AI pipeline is data collection, which involves gathering raw data from various sources [264]. This data requires extensive preprocessing to ensure quality and relevance. Techniques such as data filtering address inconsistencies and errors, resulting in a refined dataset known as cleaned data [152]. Feature extraction reduces dimensionality, producing a streamlined dataset that retains essential information while minimizing redundancy [186]. Data aggregation synthesizes information from multiple sources to enhance coherence, and data quantization lowers the precision of data representation, facilitating efficient processing on edge devices [109, 112].

2.2.2 Model Development. Once the data has been optimized, the next phase is model development [43]. This begins with model training, where algorithms learn from the cleaned and augmented data. The model design process involves selecting appropriate architectures and hyperparameters to achieve optimal performance [283]. After the initial training, model compression techniques are utilized to create a compact model that maintains accuracy while reducing computational requirements [20]. This compact model is crucial for deployment in edge environments, where resources are often limited.

2.2.3 System Integration. The final stage in the on-device AI pipeline is system integration, which encompasses both software and hardware optimization [247]. Software optimization focuses on refining the code and algorithms to enhance performance and efficiency [158]. Concurrently, hardware optimization ensures that the underlying infrastructure is capable of supporting the computational demands of the model [43]. Once these optimizations are complete, the model is deployed to edge devices, enabling real-time processing and decision-making in a variety of applications [280].

3 Applications of On-Device AI Models

The applications of on-device AI models are diverse and span various domains, including smartphones, IoT devices, edge computing, autonomous driving, and healthcare. This section provides an overview of these applications, highlighting their significance and impact.

3.1 Smartphones and Mobile Devices

Smartphones and mobile devices represent one of the most prevalent areas for the application of on-device AI models. With advancements in computational power and AI technology, smartphones can execute complex AI tasks locally. Key applications include:

- **Voice Assistants**: Devices like Apple's Siri, Google Assistant, and Amazon's Alexa utilize NLP to understand and respond to user voice commands, enhancing user interaction and accessibility [51, 144].
- **Image Recognition**: Camera applications on smartphones employ AI models for facial recognition, scene identification, and image enhancement, significantly improving photography quality [156, 263].
- **Personalized Recommendations**: By analyzing user behavior and preferences, smartphones can provide tailored app recommendations and content suggestions, thereby enhancing user engagement [184].
- **Health Monitoring**: Some smartphones are equipped with sensors that monitor health metrics such as heart rate and step count. AI models analyze this data to provide feedback and insights on user health [6].

3.2 IoT Devices

IoT devices connect via the internet to collect and exchange data, with AI models applied in various ways:

- **Smart Homes**: Devices like smart bulbs, thermostats, and security cameras use AI models for automation and intelligent decision-making. For instance, smart thermostats can automatically adjust temperatures based on user habits, enhancing energy efficiency and comfort [281].
- **Environmental Monitoring**: IoT sensors can monitor environmental data (e.g., temperature, humidity, air quality) in real-time. AI models analyze this data to provide suggestions for environmental improvements, contributing to sustainability efforts [72].
- **Industrial Automation**: In industrial IoT settings, AI models predict equipment failures, optimize production processes, and enhance efficiency while reducing maintenance costs, thereby improving overall operational reliability [107].
- **Smart Agriculture**: By collecting soil and climate data through sensors, AI models assist farmers in optimizing irrigation and fertilization practices to improve crop yields, promoting sustainable agricultural practices [147].

3.3 Edge Computing

Edge computing shifts data processing closer to the source to reduce latency and bandwidth demands. The application of AI models on edge devices includes:

- **Real-Time Data Processing**: Running AI models on edge devices enables rapid analysis of real-time data for applications such as facial recognition and behavior analysis in video surveillance, enhancing security measures [230].
- **Intelligent Traffic Management**: Edge devices can analyze traffic flow data in real-time to optimize traffic signal control and reduce congestion, improving urban mobility [115].
- **Smart Manufacturing**: Edge devices on production lines can monitor equipment status in real-time using AI models for predictive maintenance scheduling, thereby minimizing downtime and enhancing productivity [36].
- **Augmented Reality (AR) and Virtual Reality (VR)**: Edge computing supports real-time rendering and interaction for AR and VR applications, enhancing user experience and engagement in various sectors, including gaming and training [194].

3.4 Autonomous Driving and Intelligent Transportation Systems

Autonomous driving technology relies heavily on robust AI models to process data from sensors such as cameras, radar, and LiDAR. Key applications include:

- **Environmental Perception**: AI models analyze sensor data to identify surrounding objects (e.g., pedestrians, vehicles, traffic signs), ensuring safe driving and navigation [275].
- **Path Planning**: By analyzing traffic conditions and map data in real-time, AI models can plan optimal driving routes for autonomous vehicles, improving travel efficiency [102, 115].
- **Decision Making**: In complex traffic environments, AI models assist autonomous systems in making quick decisions regarding lane changes, acceleration, or deceleration, enhancing safety [62, 115].
- **Vehicle-to-Everything (V2X)**: Through communication with other vehicles and infrastructure, AI models optimize traffic flow and enhance road safety, contributing to smarter transportation systems [115, 157].

3.5 Medical Devices and Health Monitoring

The application of AI models in medical devices and health monitoring is rapidly growing:

- **Disease Diagnosis**: AI models analyze medical images (e.g., X-rays, CT scans, MRIs) to assist doctors in diagnosing diseases with improved accuracy and efficiency, facilitating timely interventions [218].
- **Personalized Treatment**: By analyzing patient health data and genetic information, AI models help physicians develop personalized treatment plans tailored to individual needs, enhancing patient outcomes [124].
- **Remote Monitoring**: Wearable devices (e.g., smartwatches) use AI models to monitor health indicators (e.g., heart rate, blood pressure) in real-time while providing health recommendations, promoting proactive health management [222].
- **Drug Development**: In drug discovery processes, AI models screen potential drug molecules to accelerate the identification and development of new medications, streamlining the research and development pipeline [154].

4 Technical Challenges of AI Models on Devices

4.1 Limited Computational Resources

AI models deployed on devices often operate in resource-constrained environments, such as smartphones, IoT devices, and edge computing nodes. These devices typically possess limited computational capabilities, which presents several challenges:

4.1.1 Processing Power. Many AI models, particularly deep learning models, require substantial computational resources for both training and inference [18]. The limited performance of CPUs and GPUs in these devices may not suffice to meet the real-time processing demands of complex models [115]. To address this challenge, optimizing algorithms to enhance computational efficiency is crucial. Techniques such as model pruning, quantization, and the use of specialized hardware accelerators can help improve processing capabilities without requiring significant increases in power consumption or hardware costs [20].

4.1.2 Model Complexity. Complex models generally demand more computational resources, resulting in increased latency during execution on edge devices, which can adversely affect user experience. Therefore, reducing model complexity while maintaining performance is a vital area of research [281]. Approaches such as designing lightweight models—like the MobileNets series [81] [183] [80]—and employing neural architecture search (NAS) techniques [25] [161] can help mitigate these issues by creating efficient models that are better suited for deployment on devices with limited resources.

4.1.3 Optimization Algorithms. To align with the computational capabilities of edge devices, researchers must develop efficient algorithms and model compression techniques. Approaches such as pruning [86], quantization [54], and knowledge distillation [268] are essential for reducing computational burdens without significantly compromising accuracy. Pruning involves removing less important weights or neurons from a model to streamline its architecture. Quantization reduces the precision of the numbers used in computations (e.g., converting floating-point weights to integers), which decreases memory usage and speeds up inference times. Knowledge distillation allows a smaller model to learn from a larger model's outputs, effectively transferring knowledge while maintaining performance. Additionally, leveraging parameter sharing [239] [167] and other optimization strategies can further enhance the efficiency of AI models on these constrained devices.

4.2 Storage and Memory Limitations

The storage and memory resources of edge devices are often limited, presenting significant challenges for the deployment and operation of AI models:

4.2.1 Storage Space. Many AI models, particularly state-of-the-art deep learning models, require substantial storage space to accommodate model parameters and intermediate results [18]. For instance, these models can demand hundreds of megabytes to over a gigabyte of storage, which often exceeds the capacity of many edge devices [18]. Therefore, effectively storing and managing models on devices with limited storage becomes a critical issue [123]. To address this challenge, model compression techniques such as pruning and quantization can significantly reduce the storage requirements of AI models [20]. Pruning involves removing less important weights or neurons from the model, while quantization reduces the precision of the weights and activations (e.g., converting floating-point numbers to integers), both of which help enable deployment on resource-constrained devices [86] [17] [279].

4.2.2 Memory Usage. Memory limitations on edge devices may hinder the ability to load all necessary data during inference, adversely affecting performance and response speed [174]. Developing memory optimization techniques is essential to reduce memory usage and enhance operational efficiency [218]. Techniques such as model distillation can be employed to create smaller, more efficient models that require less memory [20]. Additionally, incremental learning allows models to dynamically update by learning from new data without needing to store large amounts of historical data, thus retaining only the most recent information [133]. This approach not only conserves memory but also ensures that the model remains relevant and adaptive to changing user needs.

4.2.3 Data Management. In multi-user environments, effectively managing and allocating storage resources to prevent data conflicts and contention poses another challenge [107]. One potential solution is to distribute data and models across multiple edge devices, allowing them to leverage collective storage capacity and alleviate the limitations of individual devices [135]. This distributed approach can enhance resilience and scalability while improving overall system performance. Additionally, edge caching technology can be utilized to cache frequently accessed data and models between edge devices and the cloud [246]. This strategy reduces storage demands and communication costs by enabling edge devices to store commonly used data locally, minimizing the need for constant cloud access [217] [162] [180]. By implementing these strategies, organizations can optimize resource utilization while maintaining high performance in AI applications.

4.3 Energy Consumption Management

Energy consumption is a critical consideration for the operation of AI models on devices, particularly in mobile and IoT environments:

4.3.1 Battery Life. Many edge devices rely on battery power, and the high energy demands of AI models can lead to rapid battery depletion, adversely affecting user experience. Consequently, reducing the energy consumption of these models to extend battery life is an important research direction [260]. One effective approach to address this challenge is the development of energy-efficient algorithms, such as PhiNets [168, 255], which are specifically designed to minimize computational requirements and operate efficiently on devices with limited energy resources. These algorithms can help ensure that AI applications remain functional for longer periods without frequent recharging, thereby enhancing user satisfaction and device usability [286].

4.3.2 Dynamic Energy Management. To balance performance and energy use, devices must dynamically adjust their energy consumption strategies in response to varying workloads and environmental conditions [142, 240]. Researchers are focusing on developing intelligent energy management algorithms, including AI-based controllers, to optimize energy usage in edge devices [195] [205]. These dynamic management systems can monitor real-time performance metrics and adjust processing power accordingly, allowing devices to conserve energy during low-demand periods while ramping up performance when needed. This adaptability not only improves battery life but also ensures that applications run smoothly under varying conditions [286].

4.3.3 Hardware Optimization. Designing dedicated hardware accelerators, such as Tensor Processing Units (TPUs) and Field Programmable Gate Arrays (FPGAs), can significantly enhance the computational efficiency of AI models while reducing energy consumption [204] [242]. These specialized hardware solutions are optimized for specific types of computations commonly used in AI tasks, allowing for faster processing with lower power requirements compared to general-purpose processors. Additionally, advancements in energy-efficient hardware designs aim to minimize overall power usage while improving performance metrics [155] [165]. Furthermore, adopting hardware-software co-design approaches can optimize both components for energy efficiency, leading to enhanced overall system performance [90].

4.4 Communication Bandwidth Limitations

Edge devices typically face significant communication bandwidth limitations compared to servers, making it challenging to transfer large volumes of data between the edge and the cloud. This restricted connectivity poses obstacles for transmitting the substantial data required by many AI models [188]. To address these challenges and minimize communication costs, several strategies can be employed:

4.4.1 Data Preprocessing. One effective approach to reducing data transmission is through data preprocessing algorithms. These algorithms can filter and compress data, ensuring that only relevant information is transmitted during communication [231] [201]. By minimizing the amount of data that needs to be sent, these preprocessing techniques help alleviate bandwidth constraints and enhance overall communication efficiency. For instance, techniques such as feature selection and dimensionality reduction can significantly decrease the volume of data while preserving essential information, thus optimizing the transmission process [189].

4.4.2 Edge Caching. Edge caching technology is another valuable strategy that allows for the storage of frequently accessed data and models directly on edge devices [246]. By reducing the frequency of communication with the cloud, edge caching minimizes the amount of data transmitted [74]. This approach enables edge devices to quickly access locally stored information, thereby decreasing the need for cloud access and improving response times. Implementing intelligent caching strategies, such as adaptive caching based on usage patterns or predictive algorithms that anticipate future requests, can further enhance the effectiveness of edge caching solutions [1].

4.4.3 On-Device Computation. On-device computation is a further technique that facilitates real-time responses at the edge [69]. By performing computations directly on the edge device, only relevant data needs to be transmitted to the cloud, which reduces communication costs and enables faster response times [29]. This method not only

conserves bandwidth but also enhances the efficiency of data processing by allowing immediate analysis and action based on local data inputs [70]. Additionally, offloading less critical tasks to the cloud when necessary can help balance computational loads while maintaining responsiveness [47].

4.5 Data Privacy and Security

When processing sensitive data on devices, data privacy and security present significant challenges:

4.5.1 Data Protection. AI models deployed on edge devices frequently handle personal data, including health information and location data, making the security of this information during processing and storage paramount to preventing data breaches [282]. To enhance data protection in these contexts, several techniques have been proposed, such as data anonymization, trusted execution environments (TEEs), homomorphic encryption, and secure multi-party computation. Data anonymization involves removing or obfuscating personally identifiable information from datasets, ensuring individuals cannot be easily identified, which is crucial for compliance with privacy regulations [245]. TEEs create a secure area within a processor that allows sensitive data to be processed in isolation, safeguarding it from unauthorized access even if the main operating system is compromised [269] [108]. Homomorphic encryption enables computations to be performed on encrypted data without the need for decryption, thereby preserving confidentiality during processing [192] [179]. Lastly, secure multi-party computation allows multiple parties to collaboratively compute a function over their inputs while keeping those inputs private, thus facilitating secure collaborative processing [225].

4.5.2 Compliance. As data privacy regulations, such as the General Data Protection Regulation (GDPR), become increasingly stringent, AI models on edge devices must comply with relevant laws to ensure the lawful use and protection of user data [124]. Compliance is essential for maintaining user trust and avoiding legal repercussions. Organizations must implement robust data governance frameworks that include regular audits, user consent management, and transparent data handling practices to adhere to these regulations effectively [3].

4.5.3 Security Attacks. Edge devices are susceptible to a range of security threats, including malware and network attacks, prompting researchers to actively develop security mechanisms to safeguard these devices and the sensitive data they process [69]. One promising approach is federated learning, which enables AI models to be trained across a distributed network of edge devices while maintaining data privacy and security [196] [116]. By keeping the training data localized on each device and only sharing model updates, federated learning significantly reduces the risk of exposing sensitive information during the training process [257] [66]. Additionally, hybrid approaches, such as StarFL, integrate multiple strategies to tackle the unique challenges of edge computing, particularly in urban environments with high communication demands [83]. These hybrid models can dynamically adapt to varying conditions, ensuring robust security while optimizing performance.

4.6 Model Transferability and Adaptability

The transferability and adaptability of AI models on edge devices are crucial for ensuring effective operation across diverse environments:

4.6.1 Cross-Device Migration. AI models must be capable of running on various types of devices, including migrating from high-performance servers to resource-constrained mobile devices. Achieving efficient migration while maintaining performance and accuracy presents a significant challenge [115]. This involves effective model management and scheduling, which can be divided into model placement, migration, and elastic scaling [37, 212, 213]. During model placement, the first challenge is to design effective feature extraction methods that can capture relevant features from the edge environment and user tasks, given the heterogeneity of AI model requests [210]. Additionally, the complex relationships between user tasks and models, including task

dependencies, deadline restrictions, and bandwidth limitations, must be considered for optimal model placement [129] [130] [270]. Furthermore, addressing latency requirements for edge AI deployment necessitates scheduling that leverages the dependency relationships between the model's layers to minimize cold start times [127] [211] [64].

4.6.2 Environmental Adaptability. Edge devices operate in diverse environmental conditions, including variations in lighting, temperature, and network connectivity [61]. This adaptability is crucial for maintaining the performance and reliability of AI models deployed on these devices. As the operating environment changes, AI models must adjust their processing and inference strategies to ensure consistent functionality. For instance, an AI model used in a smart camera must effectively recognize objects under varying light conditions, necessitating dynamic adjustments to its algorithms to maintain accuracy and responsiveness [241]. Furthermore, temperature fluctuations can significantly affect the hardware performance of edge devices, requiring AI models that can compensate for these changes to avoid overheating or underperformance [65]. Additionally, fluctuating network conditions can impact data transmission rates; thus, AI models should be designed to function effectively with limited or intermittent connectivity [119]. This level of adaptability not only enhances the user experience but also ensures that edge devices can operate reliably in real-world scenarios where environmental conditions are unpredictable [61].

4.6.3 Continuous Learning. AI models on edge devices must possess the capability for continuous learning and updating during use to adapt to changing user needs and behavior patterns [43]. This requires models to have online learning and adaptive capabilities, enabling them to refine their performance based on real-time data and user interactions. Additionally, due to user mobility, models may need to be migrated to appropriate edge nodes to ensure optimal Quality of Service (QoS) [128]. This migration process must consider the storage structure of the AI model and the limited computing resources available in edge environments [140] [13]. Finally, to address scenarios involving sudden surges in AI model requests, effective elastic scaling strategies must be implemented. This includes accurately predicting resource utilization rates across different edge nodes and designing innovative scaling strategies that cater to the geographic distribution of users [214] [228] [134].

5 Optimization and implementation of AI models on devices

5.1 Data Optimization Techniques

In ML, the principle of "garbage in, garbage out" underscores the importance of high-quality data inputs for achieving reliable results [73]. This concept has been particularly influential in the development of large language models, where enhancements in the scale and quality of training data significantly improve model performance [264]. For effective deployment of models on edge devices, data preprocessing becomes essential [12]. This section introduces data optimization techniques commonly employed in on-device AI to ensure efficient and high-quality processing. As shown in Figure 3, these methods include data filtering, feature extraction, data aggregation, and data quantization [237], each offering specific applications and benefits tailored for on-device AI models. A summary of these techniques and their advantages is provided in Table 4, highlighting their role in enhancing the performance and efficiency of models operating in resource-constrained environments.

5.1.1 Data Filtering. Data filtering is essential for maintaining data quality by eliminating irrelevant or noisy data prior to further analysis [177]. In IoT networks, where numerous smart sensors generate vast quantities of data, the prevalence of errors and inconsistencies necessitates robust filtering techniques [200]. Active label cleaning, for instance, focuses on identifying and prioritizing visibly mislabeled data, thereby enhancing the accuracy of datasets [14]. Ensemble methods also play a significant role in effectively managing varying noise levels across datasets, ensuring that the integrity of the data is preserved during processing [152]. However, many filtering

Fig. 3. An overview of data optimization operations for on-device AI—including data filtering, feature extraction, data aggregation, data quantization, and edge computing frameworks—can be employed to enhance the quality of data collected for on-device AI models.

Table 4. Data Optimization Techniques in On-Device AI

Technique	Description and Context	Benefits and Potential Limitations
Data Filtering	Focuses on removing irrelevant or noisy data points before further processing. Commonly used in edge devices with limited memory and processing power.	**Benefits:** Enhances data quality, lowers transmission costs, and minimizes storage needs. **Limitations:** Risk of valuable data loss due to over-filtering.
Feature Extraction	Identifies and extracts relevant features from raw data to reduce dimensionality. Often applied in high-dimensional data scenarios, such as image processing.	**Benefits:** Reduces computational load, enhances interpretability, and improves model performance. **Limitations:** Possible oversimplification if critical features are missed.
Data Aggregation	Combines data from multiple sources to reduce redundancy and enhance data coherence, particularly valuable in IoT networks.	**Benefits:** Reduces transmission and storage costs, enhances data quality. **Limitations:** May introduce latency if aggregation is complex or centralized.
Data Quantization	Reduces data representation precision (e.g., from 32-bit to 8-bit) for sensor data processing in constrained environments.	**Benefits:** Decreases memory use and speeds up processing with minimal accuracy loss. **Limitations:** Balancing quantization levels can affect performance.
Edge Computing Frameworks	Leverages frameworks like AWS Greengrass or Azure IoT Edge for local data processing in real-time applications.	**Benefits:** Reduces latency and data transfer needs by enabling local processing. **Limitations:** Setup and maintenance may be resource-intensive, particularly for smaller systems.

methods can be computationally intensive, which poses challenges in resource-constrained environments typical of many IoT applications [231].

5.1.2 Feature Extraction. Feature extraction is a critical technique aimed at reducing data dimensionality, particularly important in high-dimensional contexts such as image processing [221]. By selecting a relevant subset of features, this technique retains only the essential information necessary for analysis, which minimizes model complexity and improves interpretability [221]. For example, feature selection has been shown to enhance resource efficiency in applications such as melanoma detection [46] and anomaly detection [199]. Nevertheless, there is a risk that feature extraction may oversimplify complex datasets if significant features are overlooked, potentially leading to loss of critical information [39].

5.1.3 Data Aggregation. Data aggregation involves synthesizing information from multiple sources to minimize redundancy and enhance coherence, which is particularly beneficial in IoT networks with interconnected devices [112, 262]. Techniques like federated learning enable data privacy while facilitating the combination of data from distributed sources, thus providing efficient solutions for processing large datasets [139]. However, while aggregation can improve data coherence, it may also introduce latency issues if the methods employed are overly complex or centralized [159].

5.1.4 Data Quantization. Data quantization refers to the process of reducing the precision of data representation, commonly applied in scenarios that require efficient processing of sensor data on edge devices [280]. By lowering floating-point precision, quantization significantly reduces memory usage and enhances processing speed [109]. However, careful selection of quantization levels is crucial to maintain model accuracy. For instance, sparse projection methods have demonstrated practical applications of quantization in edge environments, such as facial recognition systems [24].

5.1.5 Edge Computing Frameworks. Edge computing frameworks like AWS Greengrass and Azure IoT Edge facilitate the processing of data closer to its source, thereby reducing the need for extensive data transfer and minimizing latency in real-time applications [173]. An example includes an adaptive region-of-interest-based image compression scheme that enables rapid target detection within IoT setups [67]. Despite their advantages, these frameworks can impose substantial maintenance demands on smaller systems, which may struggle with the complexity involved [178].

5.2 Model Optimization Techniques

To effectively deploy AI models on edge devices, which often have stringent computational, memory, and power constraints, a variety of model optimization techniques have been developed [248]. These approaches aim to reduce the size and complexity of AI models while preserving their performance levels. Key techniques (as shown in Figure 4) include parameter sharing, pruning, model quantization, knowledge distillation, low-rank factorization, hardware-aware neural architecture search, and energy-efficient model design [20]. A notable example of integrating multiple optimization strategies is Deep Compression, which synergistically combines pruning, quantization, and Huffman coding to achieve substantial reductions in the size of deep neural networks (DNNs) [71]. Table 5 summarizes these techniques, detailing their descriptions, benefits, and limitations in the context of on-device AI.

5.2.1 Traditional ML Compression Methods. Before introducing the methods of DNNs, this section will first discuss traditional ML compression methods. Notable approaches include Q8KNN, which offers an 8-bit KNN quantization method for edge computing in smart lighting systems, demonstrating significant improvements in accuracy and compression ratio [176]; Stochastic Neighbor Compression, which effectively compresses datasets for k-nearest neighbor classification while enhancing robustness and speed [99]; and ProtoNN, a compressed and accurate kNN algorithm designed for resource-scarce devices, achieving excellent prediction accuracy with minimal storage and computational requirements [68]. Additionally, an efficient implementation of SVMs on low-power, low-cost 8-bit microcontrollers has been developed, enabling the deployment of smart sensors and sensor networks for intelligent data analysis, along with a new model selection algorithm tailored to fit hardware resource constraints [16]. Moreover, the ResOT model introduces resource-efficient oblique decision trees for neural signal classification, significantly reducing memory and hardware costs while maintaining classification accuracy, making it suitable for edge computing applications in medical and IoT devices [285]. Furthermore, a novel approach using memristive analog content addressable memory has been proposed to accelerate tree-based model inference, achieving substantial throughput improvements for decision tree operations [172]. Specifically, an efficient ECG classification system utilizing a resource-saving architecture and random forests has been

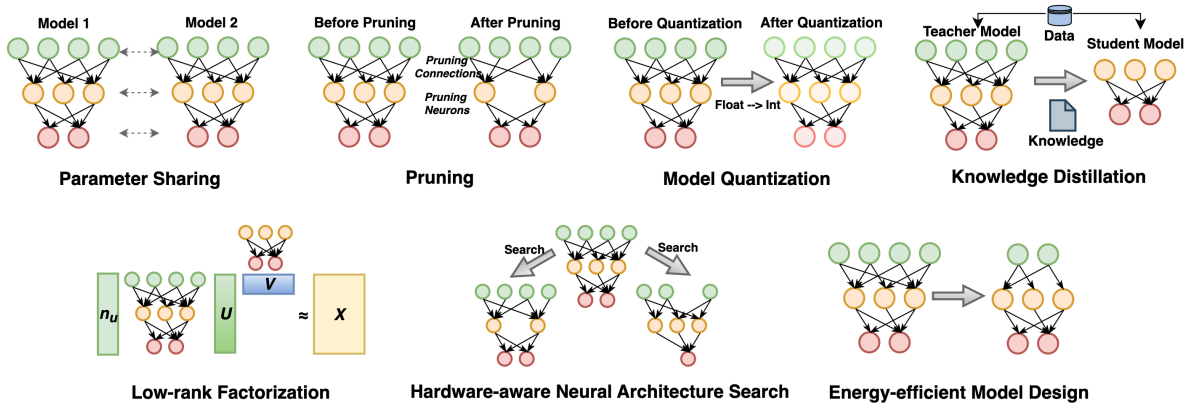

Fig. 4. An overview of model optimization operations. Model compression involves using various techniques, such as pruning, model quantization, and knowledge distillation, to reduce the size of the model and obtain a compact model that requires fewer resources while maintaining high accuracy. Model design involves creating lightweight models through manual and automated techniques, including architecture selection, parameter tuning, and regularization.

developed, achieving high classification performance for arrhythmias while maintaining low complexity and memory usage, making it suitable for wearable healthcare devices [98].

5.2.2 Parameter Sharing. Parameter sharing is a highly effective model compression technique that plays a crucial role in the development of on-device AI models [20]. By reusing weights across multiple layers, this method significantly reduces the computational and memory demands of neural networks, enabling efficient deployment on devices with limited resources without incurring substantial losses in accuracy [158]. Parameter sharing has been successfully applied across various architectures, including CNNs and RNNs, where it minimizes redundancy and optimizes memory usage—an essential requirement for on-device AI applications [167, 239]. For instance, Wu *et al.* proposed a k-means clustering approach to group weights in CNNs, allowing convolutional layers to share weights through learned cluster centers. This method effectively balances model compression with energy efficiency, making it particularly suitable for on-device applications [239]. Similarly, Obukhove *et al.* introduced T-Basis, a technique that utilizes Tensor Rings for weight compression, achieving high compression rates that are ideal for devices with limited computational power [167]. Such strategies are instrumental in on-device AI, where computational savings directly enhance device performance and prolong battery life.

However, despite its advantages, parameter sharing does present specific challenges and limitations in the context of on-device AI. While it is effective for many architectures, it may lead to decreased model interpretability and accuracy, especially if the shared parameters fail to capture task-specific nuances (Sindhwani *et al.* [191]). Additionally, the process of determining optimal shared parameters across heterogeneous operators can be computationally intensive. Techniques such as soft weight-sharing (Ullrich *et al.* [220]) and hybrid neural architecture search (You *et al.* [256]) have been developed to address these challenges, striving to balance compression with model performance through fine-grained weight sharing and architecture customization. Nonetheless, certain on-device applications, including speech recognition and recommendation systems, have successfully leveraged parameter-sharing methods to achieve efficient, high-performance models with favorable trade-offs in accuracy and resource utilization (Wang *et al.* [226]; Sun *et al.* [202]). As the demand for efficient on-device AI continues to grow, parameter sharing will remain a vital technique for optimizing model performance while accommodating the constraints of edge devices.

Table 5. Model Optimization Techniques in On-Device AI

Technique	Description	Benefits and Limitations
Parameter Sharing	Reduces the model size by sharing parameters across layers.	**Benefits**: Decreases memory usage and improves inference speed. **Limitations**: Reduces model flexibility, potentially lowering accuracy if improperly configured.
Pruning	Eliminates less important weights or entire neurons from the model.	**Benefits**: Reduces model complexity, improving execution speed without sacrificing accuracy. **Limitations**: Extensive pruning may require retraining to maintain accuracy.
Model Quantization	Lowers the precision of model weights (e.g., from 32-bit to 8-bit).	**Benefits**: Significantly lowers memory footprint and enhances performance on edge devices. **Limitations**: Can degrade model accuracy, especially with aggressive precision reductions; limited hardware compatibility.
Knowledge Distillation	Trains a smaller student model to mimic a larger, pre-trained teacher model.	**Benefits**: Achieves comparable performance with fewer parameters, ideal for resource-constrained environments. **Limitations**: Requires careful tuning and experimentation; smaller models may still underperform on complex tasks.
Low-rank Factorization	Decomposes weight matrices into lower-rank approximations to reduce model size.	**Benefits**: Maintains performance while significantly reducing the number of parameters and computational cost. **Limitations**: May require additional tuning; effectiveness can vary based on the model architecture.
Hardware-aware Neural Architecture Search	Tailors model architecture to specific hardware constraints, optimizing layers and operations.	**Benefits**: Improves efficiency by designing models that maximize hardware capabilities. **Limitations**: Computationally intensive process; may not generalize across different hardware platforms.
Energy-efficient Model Design	Focuses on minimizing energy consumption during inference.	**Benefits**: Extends battery life and improves efficiency for mobile and embedded devices. **Limitations**: Potential trade-offs in model accuracy and responsiveness.

5.2.3 Pruning. Model pruning is a vital technique for optimizing DNNs specifically for on-device AI applications, as it effectively reduces computational demands and memory usage, making models more suitable for resource-constrained edge devices [163]. By systematically removing redundant parameters or entire layers, pruning techniques decrease model complexity, enabling faster inference and lower memory consumption while often maintaining competitive accuracy [30]. Various pruning methods have been developed to enhance DNNs for on-device AI, where efficiency and generalization are critical. For instance, Xu *et al.* introduced DiReCtX, which integrates real-time pruning and accuracy tuning strategies to achieve faster model reconfiguration, improved computational performance, and significant energy savings [250]. Another notable approach is SuperSlash by Ahmad *et al.*, which employs a ranking-based pruning strategy to effectively minimize off-chip memory usage [2]. Additionally, DropNet iteratively prunes nodes or filters based on average post-activation values, achieving

up to a 90% reduction in network complexity without compromising accuracy [206]. Structural pruning methods, such as the discrete channel optimization technique developed by Gao *et al.*, yield compact models with strong discriminative power by optimizing channel-wise gates under resource constraints [56]. These advancements in pruning strategies exemplify how DNNs can be made to function efficiently on edge devices without sacrificing core functionality.

Dynamic pruning, which involves removing unimportant parameters or neurons during the training process, has also proven highly effective for on-device AI [234]. This technique is exemplified in Binarized Neural Networks (BNNs), where Geng *et al.* introduced O3BNN-R, utilizing dynamic pruning to reduce model size and energy consumption on edge devices [58]. Similarly, Li *et al.* developed FuPruner, which optimizes both parametric and nonparametric operators to accelerate neural network inference through aggressive filter pruning, achieving notable computational savings on resource-limited platforms [106]. Post-training pruning techniques have shown promise as well, with Kwon *et al.* achieving substantial reductions in computational load and inference latency for Transformers while preserving accuracy [100]. The effectiveness of pruning is further enhanced when combined with other compression techniques. For example, Lin *et al.*'s HRank utilizes low-rank feature maps for filter pruning, significantly reducing floating-point operations (FLOPs) and model size [118], while Tung *et al.*'s CLIP-Q integrates pruning with weight quantization, compressing models within a single learning framework for resource-efficient deployment [219]. These innovations underscore the importance of pruning and hybrid compression methods in enabling the deployment of high-performance neural networks on edge devices with constrained resources, ultimately facilitating more efficient on-device AI solutions.

5.2.4 Model Quantization. Quantization has emerged as a critical technique for optimizing neural networks specifically for on-device AI models, significantly enhancing computational efficiency, reducing memory and storage demands, and lowering power consumption [276]. By decreasing the precision of model parameters and activations, quantization achieves substantial reductions in model size while minimizing accuracy degradation—an essential benefit for on-device AI applications [59]. Recent advancements in quantization techniques have further improved their applicability to on-device scenarios. For instance, Fu *et al.* introduced FracTrain, which employs progressive fractional quantization and dynamic fractional quantization methods to reduce training costs and latency without sacrificing performance [54]. Similarly, Tambe *et al.* developed edgeBERT, which utilizes adaptive attention, selective pruning, and floating-point quantization to effectively address memory and computation constraints on edge devices, striking a balance between performance and resource utilization [204]. Techniques such as Parametric Non-uniform Mixed Precision Quantization have enabled data-free quantization, allowing models to be compressed without retraining—an advantageous approach for deployment on devices with limited computational capabilities [32]. For ensemble models, Cui *et al.* proposed a bit-sharing scheme that allows models to share less significant bits of parameters, optimizing memory usage while preserving accuracy [38]. These innovations reflect the growing trend of applying quantization to deploy efficient and lightweight DNNs on edge devices.

Research in quantization has also expanded to accommodate specialized neural network architectures and hardware platforms. For example, quantization frameworks tailored for Capsule Networks have been developed to address their high computational demands, achieving up to a 6.2x reduction in memory usage with minimal accuracy loss [143]. In the realm of spiking neural networks, FSpiNN incorporates fixed-point quantization to optimize memory and energy consumption for unsupervised learning on edge devices [175]. Hardware-aware quantization has gained significant traction, with systems like Zhou *et al.*'s Octo utilizing INT8 quantization to enhance cross-platform training efficiency on AI chips [279]. Additionally, Wang *et al.*'s HAQ framework applies hardware-aware quantization to select layer-specific precision levels, achieving latency and energy savings that are crucial for performance-constrained edge devices [224]. Li *et al.* introduced the RaQu framework, which

leverages resistive-memory-based processing-in-memory (RRAM-based PIM) quantization tailored for on-device AI, enhancing resource efficiency through a combined model and hardware optimization approach [103].

5.2.5 Knowledge Distillation. Knowledge distillation is a pivotal model compression technique that enhances the deployment of DNNs on resource-constrained edge devices by transferring knowledge from a large, complex teacher model to a smaller, more efficient student model [20]. Originally introduced by Hinton *et al.* [77], knowledge distillation operates by converting the teacher model's output into a softened probability distribution, which the student model learns to replicate. This approach has become foundational in enabling complex DNNs to maintain high accuracy while functioning on limited hardware [20]. A variety of knowledge distillation strategies have been developed to optimize models specifically for edge applications. For instance, Zhang *et al.* introduced a self-distillation framework that compresses knowledge within CNNs, achieving accuracy gains alongside scalable inference capabilities [268]. DynaBERT, proposed by Hou *et al.*, dynamically adjusts the width and depth of BERT models to align with the resource constraints of various edge devices, utilizing knowledge distillation to train subnetworks that perform comparably to the full model [79]. Additionally, dynamic knowledge distillation methods, such as the Dynamic Knowledge Distillation framework by Zhang *et al.*, implement adaptive features to manage sample-specific complexity, enabling deployment on devices with restricted computational power, such as satellites and UAVs [273].

Recent advancements in knowledge distillation have expanded its application to specific architectures and use cases in edge environments. For example, Hao *et al.* introduced CDFKD-MFS, which combines multiple pre-trained models into a compact student model without requiring access to the original dataset, facilitating lightweight deployment in data-restricted settings [75]. Ni *et al.* developed cross-modal Vision-to-Sensor knowledge distillation for human activity recognition, which compresses multimodal sensor data into a student model that approximates the performance of a high-complexity model while reducing computational requirements [160]. In privacy-sensitive contexts, pFedSD, a federated learning model, employs self-distillation to personalize model training for individual clients, effectively adapting to diverse edge devices and user data [92]. To further enhance model efficiency, knowledge distillation is often combined with other compression techniques. For instance, Xia *et al.* applied knowledge distillation alongside self-supervised learning in an ultra-compact recommender system, achieving significant memory savings and improved inference accuracy [243]. These innovations highlight knowledge distillation's versatility and effectiveness in adapting complex DNNs to the constraints of edge devices, achieving an optimal balance between computational cost, memory usage, and task performance. As the demand for efficient on-device AI solutions continues to grow, knowledge distillation will remain a crucial strategy for enabling high-performance models in resource-limited environments.

5.2.6 Low-rank Factorization. Low-rank factorization is a powerful technique for reducing the memory and computational requirements of DNNs, making it particularly well-suited for deployment on resource-limited edge devices [104]. By approximating weight matrices with lower-dimensional matrices, low-rank factorization captures the most significant information while minimizing redundancy [253]. One notable approach is SVD training, developed by Yang *et al.* [252], which integrates sparsity-inducing regularizers on singular values to achieve low-rank DNNs during training without the need for singular value decomposition at every step. This method effectively reduces computational load compared to prior factorization and pruning techniques while maintaining accuracy [252]. Another innovative model is MicroNet, introduced by Li *et al.* [113], which is optimized for edge devices through Micro-Factorized convolution. This approach factorizes both pointwise and depthwise convolutions to significantly reduce computational complexity. With the addition of the Dynamic Shift-Max activation function, MicroNet-M1 achieves an impressive 61.1% top-1 accuracy on ImageNet using only 12 MFLOPs, surpassing MobileNetV3's accuracy by 11.3% [113]. Despite its advantages, implementing low-rank factorization on edge devices can present challenges, particularly due to the high computational cost associated with factorization and the need for extensive retraining to achieve stable convergence [198]. Nevertheless, the

potential of low-rank factorization to enhance the efficiency of on-device AI models makes it a valuable strategy for optimizing DNNs in resource-constrained environments.

5.2.7 Hardware-aware Neural Architecture Search. Neural Architecture Search (NAS) has emerged as a crucial technique for designing neural networks tailored for edge device deployment, where constraints such as energy efficiency, low latency, and limited computational capacity are critical [137]. The rapid expansion of the IoT and AI of Things (AIoT) has created a demand for smart, low-power, and efficient devices. To meet this need, NAS employs sophisticated optimization strategies—including evolutionary algorithms, reinforcement learning, and gradient-based methods—to navigate vast architecture spaces, discovering models that excel within stringent resource limitations [33].

Recent advancements in NAS have underscored the importance of multi-objective optimization, where accuracy is balanced with key metrics like latency, memory usage, and energy consumption [131, 136]. Comparative data from various NAS frameworks highlight the trade-offs between performance and resource efficiency. For instance, MobileNetV2 achieves 72.0% accuracy with 300 million multiply-accumulate operations (MACs) and a mobile latency of 66 ms, incurring a training cost of just 150 GPU hours [183]. In contrast, NASNet-A, an early NAS model, delivers slightly higher accuracy at 74.0% but at the expense of increased computational demand—564 million MACs—and a staggering 48,000 GPU hours, leading to significant carbon emissions and training expenses [288]. Notable strides have been made with newer NAS approaches, such as DARTS and FBNet-C, which offer competitive accuracy with considerably lower training costs and environmental impact, demonstrating progress in the field [120, 238].

NAS research is increasingly focusing on hardware-aware optimization, aiming to adapt neural architectures to the specific constraints of edge devices. Techniques like ProxylessNAS exemplify this trend, balancing latency and computational needs to achieve 74.6% accuracy with just 320 million MACs and only 200 GPU hours of search cost [21]. MobileNetV3-Large, influenced by NAS principles, reaches 75.2% accuracy with reduced complexity, making it suitable for real-time edge applications [80]. Frameworks like OFA (Once-For-All) demonstrate scalability, achieving high accuracy with minimal MACs and low latency, underscoring NAS's potential to standardize efficient models across diverse platforms [19]. In federated learning, where privacy and data distribution challenges are prominent, NAS methods like Federated Direct NAS (FDNAS) and Cluster Federated Direct NAS (CFDNAS) show promise, effectively handling data heterogeneity and edge-specific requirements [265]. Collectively, these studies highlight the importance of adapting NAS to the specific challenges of edge environments, enabling the development of neural architectures that are efficient, resilient, and well-suited for real-world on-device AI deployments.

5.2.8 Energy-efficient Model Design. The design of compact neural network architectures has garnered significant attention, especially as edge devices in IoT and AIoT applications demand efficient models capable of real-time processing under stringent resource constraints [287]. Lightweight networks are specifically engineered to reduce computational demands and minimize parameter counts, making them ideal for edge platforms where power and memory are limited [20]. These models leverage strategies such as depthwise and widthwise separable convolutions, channel and network pruning, and convolutional grouping to enhance computational efficiency and reduce memory footprint without significantly sacrificing accuracy [283]. Among the most prominent lightweight models, the MobileNets series has demonstrated high performance on mobile and embedded devices by employing depth-separable convolutions that decrease computation while maintaining accuracy. For instance, MobileNetV1 achieves a 70.6% top-1 accuracy on the ImageNet dataset with only 569 million MACs, a significant reduction compared to traditional networks [81]. MobileNetV2 improves upon this by introducing inverted residuals and linear bottlenecks, achieving 72.0% top-1 accuracy with just 300 million MACs, demonstrating a substantial increase in efficiency for mobile applications [183]. MobileNetV3 further optimizes this design, achieving a 75.2% top-1 accuracy with around 219 million MACs, specifically targeting edge scenarios with

strict latency constraints [80]. Recent advancements in lightweight architecture design continue to push the boundaries of computational efficiency and accuracy. The ShuffleNet series, developed by MegVII, employs channel shuffling within grouped convolutions to optimize information flow, reducing computational costs significantly. ShuffleNetV2, for instance, achieves 72.6% top-1 accuracy on ImageNet while only requiring 299 million MACs, making it particularly effective for real-time applications on edge devices with limited processing power [141, 271]. Similarly, SqueezeNet introduces the Fire module, which compresses input channels and then expands them with fewer parameters, resulting in a model that is 50 times smaller than AlexNet while retaining comparable accuracy [88].

The EfficientNet series employs a compound scaling technique that simultaneously adjusts network width, depth, and resolution, achieving state-of-the-art accuracy while reducing parameter counts. EfficientNet-B0, for instance, achieves a top-1 accuracy of 77.3% with 5.3 billion MACs [207]. In comparison, the optimized EfficientNetV2 reaches an impressive 79.8% accuracy on the same dataset with 20% fewer parameters [208]. These models have been benchmarked on devices such as the Jetson Xavier NX and Coral Dev Board, demonstrating significant reductions in latency and energy consumption compared to traditional CNNs [110]. Additionally, attention-based lightweight architectures like MobileViT combine the strengths of CNNs and transformers to enhance global feature representation. MobileViT strikes a balance between computational efficiency and accuracy, achieving over 78% top-1 accuracy on ImageNet with approximately 320 million MACs [146]. These advancements have enabled edge devices to perform real-time inference tasks, including object detection and classification. Furthermore, Table 6 presents a comprehensive overview of lightweight models evaluated for accuracy and inference time across various mobile devices using PyTorch Mobile [132]. These models have been tested on edge platforms such as the Galaxy S10e, Honor V20, Vivo X27, Vivo Nex, and Oppo R17, highlighting their adaptability to resource-constrained environments.

Table 6. Accuracy and Inference Time for PyTorch Mobile on Different Devices [132]

Model	Galaxy S10e		Honor V20		Vivo X27		Vivo Nex		Oppo R17	
	Accuracy (%)	Time (ms)	Accuracy (%)	Time (ms)	Accuracy (%)	Time (ms)	Accuracy (%)	Time (ms)	Accuracy (%)	Time (ms)
ResNet50	74.94	333	74.94	361	74.94	1249	74.94	1243	75.16	1260
InceptionV3	77.82	433	77.82	401	77.82	1509	77.82	1500	77.68	1537
DenseNet121	74.66	246	74.66	253	74.66	915	74.66	963	74.72	982
SqueezeNet	56.94	98	56.94	105	56.94	284	56.94	284	56.74	295
MobileNetV2	70.54	269	70.54	291	70.54	769	70.54	726	70.44	734
MnasNet	72.18	289	72.18	284	72.18	694	72.18	685	72.24	735

5.3 System Optimization Techniques

As the demand for real-time performance and resource-efficient deep learning models continues to rise, optimizing systems for on-device AI deployment has become a critical area of research. Successfully deploying deep learning models on edge devices necessitates a combination of software and hardware-based approaches to enhance computational efficiency [20]. This section offers a comprehensive overview of frameworks for lightweight model training and inference from a software perspective, as well as hardware-based methods designed to accelerate model performance. Figure 5 illustrates the various system optimization approaches, highlighting how software and hardware optimizations work in tandem to improve computational efficiency for on-device AI models. This integrated strategy is essential for ensuring that deep learning applications can operate effectively within the constraints of edge environments [246].

5.3.1 Software Optimization. In on-device AI, software optimization is essential for managing and deploying lightweight models in resource-constrained environments [246]. This section categorizes software optimization approaches into two main areas: On-Device AI Learning Frameworks, which facilitate model training and

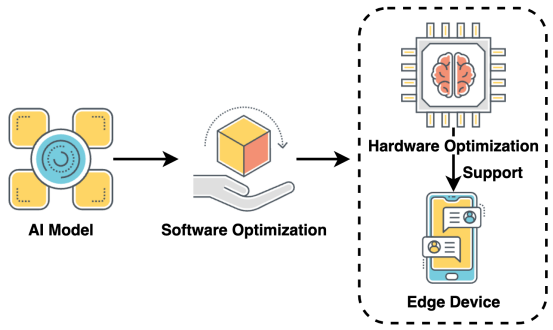

Fig. 5. An overview of system optimization operations for on-device AI. Software optimization includes frameworks for lightweight model training and inference, while hardware optimization focuses on acceleration methods to improve computational efficiency.

deployment on mobile and edge devices, and On-Device AI Inference Frameworks, which support efficient model inference across different hardware platforms.

On-Device AI Learning Frameworks: On-device AI learning frameworks are specifically designed to enable the training, optimization, and deployment of deep learning models on edge devices. Popular frameworks such as TensorFlow and PyTorch have introduced optimized versions tailored for mobile applications—namely TensorFlow Lite and PyTorch Mobile [223]. These frameworks streamline the lifecycle management of deep learning models, from training to deployment, while addressing the limitations of edge devices, including restricted computing power, memory constraints, and energy efficiency requirements [276]. A key aspect of lifecycle management in these frameworks is model conversion, which transforms complex models into lightweight versions suitable for edge deployment. For instance, TensorFlow Lite provides tools to convert standard TensorFlow models through techniques like quantization and pruning, effectively reducing model size and computational costs [235]. The conversion process typically involves exporting a trained TensorFlow model to the TensorFlow Lite Converter, which applies optimizations such as quantization to minimize resource requirements. Similarly, PyTorch Mobile enables developers to convert PyTorch models into optimized mobile versions using the TorchScript Intermediate Representation (IR), which simplifies model structures and enhances execution efficiency on mobile platforms [170]. These conversion tools not only optimize the models but also facilitate deployment on edge devices, ensuring that computationally demanding models can operate efficiently within the limited resources available. Key features of TensorFlow Lite and PyTorch Mobile are highlighted in Table 7, showcasing how each framework supports efficient model deployment in mobile and edge environments [223].

On-Device AI Inference Frameworks: On-device AI inference frameworks are specialized software environments designed to enable the efficient deployment and execution of pre-trained models on various edge devices [166]. Unlike learning frameworks, which manage the entire lifecycle from training to deployment, inference frameworks focus solely on executing models in a computationally efficient manner. Frameworks such as NCNN, OpenVINO, and ONNX Runtime are tailored for edge applications, providing optimized implementations that reduce memory and power consumption while supporting a range of hardware platforms, including IoT devices, mobile phones, and edge servers [89, 151, 216]. These frameworks integrate performance optimizations specific to common architectures and operations, such as quantization and low-precision computation, to facilitate high-speed, low-latency model inference on devices with limited computational resources. For example, ONNX Runtime [151] offers significant speedups for inference and training across diverse platforms, while OpenVINO [89] optimizes

Table 7. On-Device AI Learning Frameworks

Framework	Producer	Highlights
TensorFlow Lite	Google	• Optimizes on-device ML by addressing latency, privacy, connectivity, size, and power consumption • Supports multiple platforms (e.g., Android, iOS, embedded Linux, microcontrollers) • Multiple programming languages supported, including Java, Swift, Objective-C, C++, and Python • High-performance features such as hardware acceleration and model optimization • Prebuilt examples for common ML tasks across multiple platforms
Pytorch Mobile	Facebook	• Compatible with iOS, Android, and Linux platforms • Provides APIs for preprocessing and model integration tasks • Supports TorchScript IR for model tracing and scripting • Offers XNNPACK floating point and QNNPACK 8-bit quantized kernels for ARM CPUs • Features an optimized mobile interpreter and streamlined model optimization through optimize_for_mobile

deep learning models for Intel hardware, incorporating functions like FP16 and INT8 quantization to enhance throughput. NCNN, developed by Tencent, emphasizes minimal memory usage and compatibility with ARM processors, making it particularly suitable for mobile deployments [216]. Other frameworks, such as Arm NN [8] and MNN [4], are similarly designed for cross-platform deployment, supporting a variety of model types and hardware backends. Table 8 summarizes the key attributes, supported hardware, advantages, and limitations of various on-device AI inference frameworks.

Recent advancements in on-device AI have led to the development of numerous lightweight neural network architectures and frameworks, particularly for CNNs. For instance, SparkNet, introduced by Xia *et al.* [242], reduces model parameters and computational requirements for CNNs, achieving high efficiency in resource-constrained environments. Memsqueezer, developed by Wang *et al.* [233], utilizes on-chip memory architectures to optimize CNN inference, resulting in a 2x performance improvement and an 80% reduction in energy consumption. Other frameworks, such as Pipe-it [227] and SCA [274], implement techniques like kernel parallelization and secure computation, respectively, to boost CNN inference throughput and protect model integrity, demonstrating significant improvements in speed, latency, and energy consumption across diverse hardware configurations. For RNNs and other neural network architectures, compression and pruning techniques have been central to enabling efficient edge deployment. Gao *et al.* [55] introduced EdgeDRNN, which leverages temporal sparsity to enhance power efficiency and reduce latency for RNN inference, making it suitable for real-time applications. Compression-based approaches, such as those by Srivastava *et al.* [197] and Wen *et al.* [236], employ variational bottleneck and structured pruning, respectively, to significantly reduce RNN model size and memory footprint while preserving performance. Zhang *et al.* [267] proposed DirNet, an adaptive compression method that adjusts sparsity in RNNs, allowing deployment on resource-limited edge devices without sacrificing accuracy. Additional developments in edge-optimized deep neural networks, such as Hidet [45] and edgeEye [121], have further enhanced the scope and efficiency of AI inference on mobile platforms, enabling real-time video analytics and other demanding applications in constrained environments.

5.3.2 Hardware Optimization. Hardware-based optimization methods enhance computational efficiency for on-device AI models by utilizing specialized hardware accelerators and low-power chips [20]. These methods include compression algorithms, memory-efficient architectures, and domain-specific hardware, enabling high-performance AI processing on devices with limited power budgets [190]. Various approaches to hardware acceleration include using specialized processors, such as CPUs, GPUs, FPGAs, ASICs, and NPUs, or implementing custom hardware designs for specific AI models [42, 171]. Each of these hardware options provides unique benefits [52]: CPUs are versatile and have a stable computing performance; GPUs offer high parallelism and flexibility; FPGAs allow for customizability and low power consumption; ASICs achieve high efficiency through hardware

Table 8. On-Device AI Inference Frameworks

Framework	Producer	Supported Hardware	Advantages	Limitations
ONNX Runtime [151]	Microsoft	CPU, GPU, etc	• It has built-in optimizations that can boost inferencing speed up to 17 times and training speed up to 1.4 times • It supports multiple frameworks, operating systems, and hardware platforms • High performance, and low latency	• Limited support for non-ONNX models • No support for some hardware backends
OpenVINO [89]	Intel	CPU, GPU, VPU, FPGA, etc	• It optimizes deep learning pipelines for high performance and throughput • Support for advanced functions such as FP16, INT8 quantization • It supports multiple deep learning frameworks and multiple operating systems	• Only Intel hardware products are supported • Deploying and integrating models still requires some technical knowledge and experience
NCNN [216]	Tencent	CPU, GPU, etc	• High performance and low memory usage • Supports a variety of hardware devices and model formats • Supports 8-bit quantization and ARM NEON optimization	• Limited support for non-NCNN models
Arm NN [8]	Arm	CPU, GPU, etc	• Cross platform • Supports a variety of hardware devices and model formats • Existing software can automatically take advantage of new hardware features • Support for ARM Compute Library	• Limited support for operators and network structures
MNN [4]	Alibaba	CPU, GPU, NPU	• MNN is a lightweight, device-optimized framework with quantization support • MNN is versatile, supporting various neural networks and models, multiple inputs/outputs, and hybrid computing on multiple devices • MNN achieves high performance through optimized assembly, GPU inference, and efficient convolution algorithms. • MNN is easy to use, with support for numerical calculation, image processing, and Python API	• Limited community support • Technical expertise required
TensorRT [166]	NVIDIA	CPU, GPU	• Maximize throughput by quantifying the model to INT8 while maintaining high accuracy • Optimize GPU video memory and bandwidth usage by merging nodes in the kernel • Select the best data layer and algorithm based on the target GPU platform • Minimize video memory footprint and efficiently reuses memory for tensors • An extensible design for processing multiple input streams in parallel	• It only runs on NVIDIA graphics cards • It does not open source the kernel
TVM [7]	Apache	CPU, GPU, DSP, etc	• Compilation and minimal runtimes optimize ML workloads on existing hardware for better performance • Supports a variety of hardware devices and model formats • TVM's design enables flexibility for block sparsity, quantization, classical ML, memory planning, etc	• Deploying and integrating models still requires some technical knowledge and experience

optimization; and NPUs are tailored specifically for deep learning. Table 9 provides an overview of common hardware accelerators for edge AI, including their basic information, examples, advantages, and limitations.

Table 9. On-Device AI Model Accelerators

Hardware	Basic Information	Examples	Advantages	Limitations
CPU	General-purpose computing hardware suitable for diverse applications	• ARM Cortex-M55 • Intel Atom x7-E3950 • Qualcomm Snapdragon 865 • Apple A14 Bionic • MediaTek Helio P90	• Versatile across multiple application scenarios • Reliable and stable performance • Broad support from established hardware and software ecosystems	• Limited computational power for AI compared to GPUs and ASICs • Less efficient in power-sensitive environments requiring high performance
GPU	Specialized hardware for accelerating deep learning and parallel computing tasks	• Microsoft Azure Stack Edge • Lenovo ThinkEdge SE50 • NVIDIA Jetson Xavier NX • Raspberry Pi 4	• High parallel processing capabilities • Flexible for various AI workloads • Extensive support for AI applications	• Higher power consumption, potentially unsuitable for low-power edge devices
FPGA	Customizable hardware to accelerate deep learning tasks through programmable logic	• Lattice sensAI • QuickLogic EOS S3 • Xilinx Zynq UltraScale+ • Intel Movidius Neural Compute Stick	• High flexibility for custom AI applications • Energy efficient with low power consumption • Tailored hardware solutions for specific tasks	• Higher complexity in development and deployment • Longer time-to-market compared to off-the-shelf solutions
ASIC	Optimized hardware for achieving maximum performance and energy efficiency in AI tasks	• Google Edge TPU • Horizon Robotics Sunrise • MediaTek NeuroPilot • Cambricon MLU100	• Hardware tailored for specific AI workloads • Exceptional energy efficiency with low power requirements • Superior performance due to optimized architecture	• High initial development and manufacturing costs • Lack of flexibility for different or updated AI models
NPU	Specialized processors designed for efficient deep learning acceleration	• Qualcomm Hexagon 680 • Apple Neural Engine • Huawei Kirin 990 • MediaTek APU • NVIDIA Jetson Nano	• High efficiency in processing neural network tasks • Optimized for low power consumption • Provides dedicated hardware for AI inference	• Limited compatibility with certain AI models and frameworks • May not support highly customized or complex deep learning tasks

In recent years, the development of on-device AI models has gained significant traction, particularly in edge AI applications, where resource constraints and power efficiency are critical. CPU-based accelerators have emerged as a viable solution due to their broad applicability and stable performance. For instance, Nori *et al.* [164] introduced REDUCT, a DNN inference framework that optimizes data-parallel processing on multi-core CPUs, bypassing traditional CPU resources. This approach resulted in a 2.3x increase in convolution performance per watt, demonstrating substantial gains in both raw performance and power efficiency for on-device AI tasks. Similarly, Zhu *et al.* [91] developed NCPU, a neural CPU architecture that integrates a binary neural network accelerator with a RISC-V CPU pipeline. NCPU's support for local data storage minimizes costly data transfers between cores, leading to significant area reduction and energy savings compared to conventional architectures. These advancements illustrate the potential of CPUs to effectively meet the demands of on-device AI by maximizing resource utilization while minimizing power consumption.

GPUs are also widely utilized for accelerating deep learning tasks on edge devices, thanks to their parallel processing capabilities. Capodieci *et al.* [22] showcased a real-time scheduling prototype for NVIDIA GPUs, incorporating preemptive scheduling and bandwidth isolation techniques that enhance performance for repeated tasks in deep learning applications. This capability is crucial for on-device AI models that require efficient resource management. FPGAs offer another effective approach to deep learning acceleration on edge devices.

Table 10. Summary of On-Device AI Model Accelerator from the Literature

Method	Hardware	Model	Strategy	Performance
REDUCT [164]	CPU	DNN	• It bypasses CPU resources to optimize DNN inference	• 2.3x increase in convolution performance/Watt • 2x to 3.94x scaling in raw performance • 1.8x increase in inner-product performance/Watt • 2.8x scaling in performance
NCPU [91]	CPU	BNN	• Propose a unified architecture	• Achieved 35% area reduction and 12% energy saving compared to conventional heterogeneous architecture • Implemented two-core NCPU SoC achieves an end-to-end performance speed-up of 43% or an equivalent 74% energy saving
Prototype [22]	GPU	DNN	• The schedulability of repeated real-time GPU tasks is significantly improved	• Achieved 35% area reduction and 12% energy saving compared to conventional heterogeneous architecture • Implemented two-core NCPU SoC achieves an end-to-end performance speed-up of 43% or an equivalent 74% energy saving
SparkNoC [242]	FPGA	CNN	• Simultaneous pipelined work	• Performance: 337.2 GOP/s • Energy efficiency: 44.48 GOP/s/w
FPGA Overlay [35]	FPGA	CNN	• It exploits all forms of parallelism inside a convolution operation	• An improvement of 1.2x to 5x in maximum throughput • An improvement of 1.3x to 4x in performance density
Light-OPU [258]	FPGA	CNN	• With a corresponding compilation flow	• Achievement of 5.5x better latency and 3.0x higher power efficiency on average compared with NVIDIA Jetson TX2 • Achievement of 1.3x to 8.4x better power efficiency compared with previous customized FPGA accelerators
edgeBert [204]	ASIC	Transformer	• It employs entropy-based early exit predication	• The energy savings are up to 7x, 2.5x, and 53x compared to conventional inference without early stopping, latency-unbounded early exit approach
ApGAN [181]	ASIC	GAN	• By binarizing weights and using a hardware-configurable in-memory addition scheme	• Achieve energy efficiency improvements of up to 28.6x • Achieve a 35-fold speedup
Fluid Batching [97]	NPU	DNN	• Fluid Batching and Stackable Processing Elements are introduced	• 1.97x improvement in average latency • 6.7x improvement in tail latency SLO satisfaction
BitSystolic [254]	NPU	DNN	• Based on a systolic array structure	• It achieves high power efficiency of up to 26.7 TOPS/W with 17.8 mW peak power consumption
PL-NPU [232]	NPU	DNN	• A posit-based logarithm-domain processing element, a reconfigurable inter-intra-channel-reuse dataflow, and a pointed-stake-shaped codec unit are employed	• 3.75x higher energy efficiency • 1.68x speedup
FARNN [34]	FPGA + GPU	RNN	• To separate RNN computations into different tasks that are suitable for GPU or FPGA	• Improve by up to 4.2x
DART [244]	CPU + GPU	DNN	• It offers deterministic response time to real-time tasks and increased throughput to best-effort tasks	• Response time was reduced by up to 98.5% • Achieve up to 17.9% higher throughput

Xia *et al.* [242] introduced an FPGA-based architecture optimized for SparkNet, achieving high performance and energy efficiency through a fully pipelined CNN accelerator. Choudhury *et al.* [35] further proposed an FPGA overlay optimized for CNN processing, exploiting parallelism to maximize throughput based on available compute and memory resources. Additionally, Yu *et al.* [258] developed a lightweight FPGA overlay processor for CNNs, utilizing a specialized compilation flow to achieve 5.5x better latency and 3.0x higher power efficiency than the NVIDIA Jetson TX2. These developments highlight how FPGAs can be tailored to optimize deep learning performance for on-device AI applications, providing customizable and scalable acceleration.

ASICs are increasingly favored for on-device AI due to their hardware-level optimization and energy efficiency. For example, Tambe *et al.* [204] designed edgeBERT, an ASIC-based architecture for multi-task NLP inference, which employs an entropy-based early exit mechanism to achieve energy savings of up to 7x compared to conventional inference methods. Roohi *et al.* [181] introduced ApGAN, leveraging a binarized weight approach and a hardware-configurable addition scheme for GANs, resulting in energy efficiency improvements of up to 28.6x. NPUs, specialized ASICs designed for neural network processing, also play a crucial role in on-device AI by offering high efficiency and low power usage. Kouris *et al.* [97] presented Fluid Batching, a novel NPU architecture that enhances utilization and improves latency. Other architectures, such as BitSystolic [254] and PL-NPU [232], provide significant power efficiency and speed improvements for DNN inference through mixed-precision arithmetic and dataflow optimization techniques, respectively. Some solutions combine different processors, such as FPGA + GPU [34] and CPU + GPU [244], to balance performance and efficiency by leveraging the strengths of each processor. Overall, ASICs and NPUs, with their low power consumption and optimized structures, represent highly effective solutions for accelerating on-device AI models, particularly in power-constrained environments. Specifically, Table 10 summarizes the hardware, models, strategies, and performance metrics of these hardware optimization techniques, highlighting their relevance to the advancement of on-device AI.

6 Future Development Trends

6.1 Impact of Emerging Technologies

The rapid advancement of emerging technologies is poised to significantly influence the application and performance of AI models deployed on various devices. Key technologies such as 5G, edge computing, and foundation models will play critical roles in shaping the future landscape of AI:

6.1.1 5G and Beyond. The rollout of more advanced networks like 5G, characterized by high bandwidth and low latency, is set to enhance the real-time processing capabilities of AI models on devices [101]. With this connectivity, devices can access cloud resources more swiftly, facilitating seamless data exchange and model updates [194]. This improvement will lead to enhanced responsiveness in intelligent applications, enabling scenarios such as real-time video processing and instant feedback in smart devices. Moreover, more advanced communication technology will support the development of edge computing by allowing data processing to occur closer to the data source [125]. This proximity reduces latency and enhances data security, as edge devices can analyze data in real time [278]. Applications such as smart transportation systems, smart cities, and industrial automation stand to benefit from these advancements, resulting in more efficient operations and improved user experiences [117].

6.1.2 Edge Computing. Edge computing is pivotal for enabling AI models to process data closer to its source, thereby reducing reliance on centralized cloud computing [185]. By executing AI models on edge devices, organizations can achieve faster decision-making and responses, which are essential for applications requiring real-time feedback, such as autonomous driving and smart surveillance systems [43]. Additionally, edge computing alleviates bandwidth requirements by minimizing the amount of data transmitted to the cloud [188]. This reduction

not only lowers data transmission costs but also enhances data privacy protection by keeping sensitive information local [284].

6.1.3 Foundation Models. Foundation models represent a significant leap in AI technology due to their ability to be pre-trained on extensive datasets and subsequently fine-tuned for specific tasks [251]. These versatile models serve as a robust base for various applications, enabling faster development and deployment of AI solutions across multiple domains [259]. Their adaptability allows them to be effectively utilized in edge computing environments, where they can be tailored to meet local application needs while leveraging the extensive knowledge encoded during their pre-training phase [248]. This capability enhances both the efficiency and performance of AI models deployed on edge devices, making them more responsive to user demands and environmental changes [48].

6.2 Adaptability and Intelligence of AI Models

Future AI models are expected to exhibit greater adaptability and intelligence to meet evolving environmental conditions and user requirements:

6.2.1 Adaptive Learning. AI models will increasingly incorporate adaptive capabilities that allow them to dynamically adjust based on real-time data inputs and user feedback [229]. This adaptability is crucial for maintaining high performance across diverse environments and conditions [87]. For instance, smart home devices can automatically modify their settings according to users' habits, thereby providing personalized services that enhance user satisfaction [41]. Such adaptive learning mechanisms not only improve user experience but also optimize the functionality of devices in varying contexts, ensuring that they remain relevant and effective as conditions change [126].

6.2.2 Intelligent Decision-Making. Future AI models will be designed to make more complex decisions by integrating multiple data sources along with contextual information [43]. This integration will facilitate more accurate predictions and recommendations, allowing AI systems to function more intelligently in real-world applications [284]. By leveraging techniques from reinforcement learning and deep learning, these models will be capable of autonomously learning from their environments and optimizing their performance over time [27, 209]. This capability to process complex datasets and derive meaningful insights will significantly elevate the overall intelligence of AI systems, making them more effective in tasks ranging from autonomous navigation to personalized healthcare solutions [76, 94].

6.3 Sustainability and Green Computing of AI Models on Devices

With a growing emphasis on environmental protection and sustainable development, the sustainability and green computing aspects of AI models deployed on devices are becoming increasingly important:

6.3.1 Energy Efficiency Optimization. Future AI models will prioritize energy efficiency by minimizing energy consumption through optimized algorithms and hardware design [287]. The adoption of low-power hardware, combined with efficient computational methods, will significantly contribute to achieving green computing objectives [142]. For instance, techniques such as model pruning, quantization, and knowledge distillation can reduce the computational load of AI models, allowing them to operate effectively on devices with limited power resources [246]. Furthermore, the integration of energy-efficient architectures such as neuromorphic computing can lead to substantial reductions in power consumption while maintaining high performance levels [43].

6.3.2 Resource Sharing and Circular Utilization. AI models operating on devices will promote resource sharing and circular utilization practices that minimize resource waste [111]. Collaborative cloud-edge computing architectures allow devices to dynamically access cloud resources as needed, optimizing overall resource utilization efficiency [49, 114]. This approach not only enhances the computational capabilities of edge devices but also reduces the

environmental impact associated with over-provisioning resources [272]. By enabling devices to share processing tasks with cloud resources during peak loads or when additional capacity is required, organizations can achieve a more sustainable operational model that aligns with circular economy principles [63].

6.3.3 Environmental Monitoring and Management. AI models will play a crucial role in environmental monitoring and management by analyzing environmental data to support sustainable development goals [15]. For example, AI systems can process data from various sensors to monitor air quality, water usage, and energy consumption in real time [84]. This capability enables organizations to identify inefficiencies and implement corrective actions promptly, thereby reducing carbon emissions and optimizing resource management practices [138]. Moreover, AI-driven predictive analytics can forecast environmental changes, helping policymakers make informed decisions that contribute to sustainability efforts [142].

6.4 Ethics and Social Impact

As AI technology becomes more pervasive, addressing ethical considerations and social impacts will be critical issues that cannot be overlooked:

6.4.1 Data Privacy and Security. When processing user data on devices, safeguarding user privacy and ensuring data security remain significant challenges [5]. Future AI models must comply with stringent data protection regulations, such as the General Data Protection Regulation, to ensure the safety of user information [148]. Developing transparent algorithms alongside clear data usage policies is essential for enhancing user trust in AI technologies. This transparency can be achieved through explainable AI frameworks, which allow users to understand how their data is being used and how decisions are made by AI systems [85, 95]. Additionally, implementing robust encryption methods and secure data storage practices will further protect sensitive information from unauthorized access and breaches [192] [179].

6.4.2 Fairness and Bias. AI models may inadvertently introduce biases derived from training datasets, leading to unfair decision-making outcomes [182]. This bias can manifest in various forms, such as racial, gender, or socioeconomic biases, which can perpetuate inequality in critical areas like hiring practices, law enforcement, and loan approvals [145]. Future research should focus on identifying and eliminating biases within these models to ensure fairness in AI technologies. Techniques such as bias detection algorithms and fairness-aware ML can help mitigate these issues [187]. Establishing evaluation standards alongside regulatory mechanisms is necessary for ensuring transparency and interpretability within AI systems [145]. Furthermore, involving diverse stakeholders in the development process can help identify potential biases early on and promote equitable outcomes [28].

6.4.3 Social Impact. The widespread adoption of AI technology is expected to have profound effects on employment dynamics, educational structures, and social frameworks [53]. As automation increases, certain job categories may diminish while new roles emerge that require advanced skills in technology management and AI oversight [82]. Attention must be directed towards understanding how AI influences labor markets while promoting human-machine collaboration aimed at enhancing human skills [93]. Upskilling initiatives and educational programs will be essential in preparing the workforce for the changes brought about by AI integration. Policymakers must work collaboratively with researchers to ensure that the sustainable development of AI technology benefits society at large while addressing potential disruptions in employment [149].

7 Conclusion

7.1 Main Findings of the Survey

This survey investigates the fundamental concepts, application scenarios, technical challenges, and optimization and implementation methods associated with AI models deployed on devices. The key findings are summarized as follows:

- **Diverse Application Scenarios**: AI models on devices exhibit extensive application potential across a multitude of domains, including smartphones, IoT devices, edge computing, autonomous driving, and medical devices. This versatility significantly contributes to the proliferation and advancement of intelligent technologies [82, 96, 122].
- **Technical Challenges**: Despite the considerable advantages offered by AI models on devices, several challenges persist. These include limitations in computational resources, constraints related to storage and memory, energy management issues, as well as concerns regarding data privacy and security, alongside challenges related to model transferability and adaptability [11, 20, 284].
- **Advancements in Optimization Technologies**: The performance of AI models on devices has been markedly improved through the application of various optimization techniques, such as model compression, pruning, hardware acceleration, quantization, low-precision computing, and methodologies like transfer learning and federated learning. These advancements enable efficient operation within resource-constrained environments [20, 105, 190].
- **Future Development Trends**: The emergence of new technologies is poised to further propel the development of AI models on devices, enhancing their adaptive and intelligent capabilities. Concurrently, sustainability and ethical considerations are expected to gain prominence as critical focal points in this evolving landscape [44, 248].

7.2 Recommendations for Future Research

Building on the insights gleaned from this survey, it is essential to identify key areas for future research that can further advance the field of AI models on devices. The following recommendations aim to address existing challenges, enhance the effectiveness of AI implementations, and ensure that these technologies are developed responsibly and sustainably:

- **Research on Optimization Algorithms**: There is a pressing need to continue exploring efficient model compression and pruning techniques aimed at further reducing the computational and storage demands of AI models on devices, all while preserving accuracy and performance [20, 27, 158].
- **Co-design of Hardware and Software**: Future investigations should focus on optimizing the integration of AI models with emerging hardware technologies, such as FPGAs and TPUs, to achieve enhanced computational efficiency and improved energy management [43, 90, 284].
- **Data Privacy and Security**: It is essential to develop more robust data protection mechanisms and privacy-preserving algorithms to ensure the security and compliance of sensitive data processing on devices [5, 82, 284].
- **Fairness and Explainability**: Strengthening research efforts focused on the fairness and explainability of AI models is crucial to prevent the introduction of biases in decision-making processes, thereby enhancing user trust in AI technologies [23, 85, 95].
- **Interdisciplinary Collaboration**: Promoting interdisciplinary collaboration among fields such as computer science, social sciences, and ethics is vital for comprehensively understanding the societal impacts of AI technologies and for formulating appropriate policies and standards [9, 78].

7.3 Potential Impacts and Prospects of AI Models on Devices

The widespread integration of AI models into various devices is expected to bring about significant transformations across multiple dimensions, including societal, economic, and technological realms. As these intelligent systems become increasingly embedded in everyday life, their influence will extend beyond mere convenience, fundamentally reshaping how individuals interact with technology and each other. The following points outline the anticipated impacts and future prospects of AI models on devices:

- **Improving Quality of Life**: Through innovations in smart homes, health monitoring, and personalized services, AI models on devices are set to significantly enhance individuals' quality of life, facilitating more convenient and efficient lifestyles [70, 135].
- **Driving Industrial Transformation**: In sectors such as manufacturing, transportation, and healthcare, the deployment of AI models on devices will catalyze the intelligent transformation of industries, leading to improved productivity and service quality, and ultimately fostering economic growth [11, 177].
- **Promoting Sustainable Development**: By optimizing resource utilization and enhancing environmental monitoring, AI models on devices will play a pivotal role in achieving sustainable development goals, addressing pressing global challenges such as climate change and resource scarcity [142, 286].
- **Triggering Social Change**: The proliferation of AI technologies is expected to reshape labor markets, foster human-machine collaboration, and enhance human skills and capabilities. However, this transformation will also necessitate careful consideration of potential social inequalities and ethical issues that may arise as a result of these advancements [11, 82].

The development prospects of AI models on devices are vast and promising. Through continuous technological innovation and research, AI models on devices will bring transformative changes across various industries, advancing the process of societal intelligence. Future research and practice should focus on the sustainability and ethical implications of technology to ensure the healthy development of AI technologies for the benefit of all humanity.

Acknowledgments

This work was supported in part by the Chinese National Research Fund (NSFC) under Grant 62272050 and Grant 62302048; in part by the Guangdong Key Lab of AI and Multi-modal Data Processing, United International College (UIC), Zhuhai under 2023-2024 Grants sponsored by Guangdong Provincial Department of Education; in part by Institute of Artificial Intelligence and Future Networks (BNU-Zhuhai) and Engineering Center of AI and Future Education, Guangdong Provincial Department of Science and Technology, China; Zhuhai Science-Tech Innovation Bureau under Grant No. 2320004002772, and in part by the Interdisciplinary Intelligence SuperComputer Center of Beijing Normal University (Zhuhai).

References

[1] Rafat Aghazadeh, Ali Shahidinejad, and Mostafa Ghobaei-Arani. 2023. Proactive content caching in edge computing environment: A review. *Software: Practice and Experience* 53, 3 (2023), 811–855.

[2] Hazoor Ahmad, Tabasher Arif, Muhammad Abdullah Hanif, Rehan Hafiz, and Muhammad Shafique. 2020. SuperSlash: A unified design space exploration and model compression methodology for design of deep learning accelerators with reduced off-chip memory access volume. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, 11 (2020), 4191–4204.

[3] Mohammad Al-Rubaie and J Morris Chang. 2019. Privacy-preserving machine learning: Threats and solutions. *IEEE Security & Privacy* 17, 2 (2019), 49–58.

[4] Alibaba. 2018. MNN is a blazing fast, lightweight deep learning framework https://github.com/alibaba/MNN. *GitHub repository* (2018).

[5] Abdulmalik Alwarafy, Khaled A Al-Thelaya, Mohamed Abdallah, Jens Schneider, and Mounir Hamdi. 2020. A survey on security and privacy issues in edge-computing-assisted internet of things. *IEEE Internet of Things Journal* 8, 6 (2020), 4004–4022.

[6] Chioma Virginia Anikwe, Henry Friday Nweke, Anayo Chukwu Ikegwu, Chukwunonso Adolphus Egwuonwu, Fergus Uchenna Onu, Uzoma Rita Alo, and Ying Wah Teh. 2022. Mobile and wearable sensors for data-driven health monitoring system: State-of-the-art and future prospect. *Expert Systems with Applications* 202 (2022), 117362.

[7] Apache. 2018. Open deep learning compiler stack for cpu, gpu and specialized accelerators. https://github.com/apache/tvm. (2018).

[8] Arm. 2018. Arm NN ML Software. https://github.com/ARM-software/armnn. *GitHub repository* (2018).

[9] Jokubas Ausra, Micah Madrid, Rose T Yin, Jessica Hanna, Suzanne Arnott, Jaclyn A Brennan, Roberto Peralta, David Clausen, Jakob A Bakall, Igor R Efimov, et al. 2022. Wireless, fully implantable cardiac stimulation and recording with on-device computation for closed-loop pacing and defibrillation. *Science advances* 8, 43 (2022), eabq7469.

[10] Emna Baccour, Naram Mhaisen, Alaa Awad Abdellatif, Aiman Erbad, Amr Mohamed, Mounir Hamdi, and Mohsen Guizani. 2022. Pervasive AI for IoT applications: A survey on resource-efficient distributed artificial intelligence. *IEEE Communications Surveys & Tutorials* 24, 4 (2022), 2366–2418.

[11] Chunguang Bai, Patrick Dallasega, Guido Orzes, and Joseph Sarkis. 2020. Industry 4.0 technologies assessment: A sustainability perspective. *International Journal of Production Economics* 229 (2020), 107776.

[12] Mohammed Djameleddine Belgoumri, Mohamed Reda Bouadjenek, Sunil Aryal, and Hakim Hacid. 2024. Data Quality in Edge Machine Learning: A State-of-the-Art Survey. *arXiv preprint arXiv:2406.02600* (2024).

[13] Thad Benjaponpitak, Meatasit Karakate, and Kunwadee Sripanidkulchai. 2020. Enabling live migration of containerized applications across clouds. In *In Proceedings of 2020 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2529–2538.

[14] Mélanie Bernhardt, Daniel C Castro, Ryutaro Tanno, Anton Schwaighofer, Kerem C Tezcan, Miguel Monteiro, Shruthi Bannur, Lungren, et al. 2022. Active label cleaning for improved dataset quality under resource constraints. *Nature Communications* 13, 1 (2022), 1161.

[15] Simon Elias Bibri, John Krogstie, Amin Kaboli, and Alexandre Alahi. 2024. Smarter eco-cities and their leading-edge artificial intelligence of things solutions for environmental sustainability: A comprehensive systematic review. *Environmental Science and Ecotechnology* 19 (2024), 100330.

[16] Andrea Boni, Fernando Pianegiani, and Dario Petri. 2007. Low-power and low-cost implementation of SVMs for smart sensors. *IEEE Transactions on Instrumentation and Measurement* 56, 1 (2007), 39–44.

[17] Yoonho Boo, Sungho Shin, Jungwook Choi, and Wonyong Sung. 2021. Stochastic precision ensemble: self-knowledge distillation for quantized deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 6794–6802.

[18] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Shyam, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems* 33 (2020), 1877–1901.

[19] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. 2019. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791* (2019).

[20] Han Cai, Ji Lin, Yujun Lin, Zhijian Liu, Haotian Tang, Hanrui Wang, Ligeng Zhu, and Song Han. 2022. Enable deep learning on mobile devices: Methods, systems, and applications. *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 27, 3 (2022), 1–50.

[21] Han Cai, Ligeng Zhu, and Song Han. 2018. Proxylessnas: Direct neural architecture search on target task and hardware. *arXiv preprint arXiv:1812.00332* (2018).

[22] Nicola Capodieci, Roberto Cavicchioli, Marko Bertogna, and Aingara Paramakuru. 2018. Deadline-based scheduling for GPU with preemption support. In *2018 IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 119–130.

[23] Bhanu Chander, Chinju John, Lekha Warrier, and Kumaravelan Gopalakrishnan. 2024. Toward trustworthy artificial intelligence (TAI) in the context of explainability and robustness. *Comput. Surveys* (2024).

[24] Dong Chen, Xudong Cao, Fang Wen, and Jian Sun. 2013. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3025–3032.

[25] Hanlin Chen, Li'an Zhuo, Baochang Zhang, Xiawu Zheng, Jianzhuang Liu, Rongrong Ji, David Doermann, and Guodong Guo. 2021. Binarized neural architecture search for efficient object recognition. *International Journal of Computer Vision* 129 (2021), 501–516.

[26] Jianguo Chen, Kenli Li, Zhaolei Zhang, Keqin Li, and Philip S Yu. 2021. A survey on applications of artificial intelligence in fighting against COVID-19. *ACM Computing Surveys (CSUR)* 54, 8 (2021), 1–32.

[27] Jiasi Chen and Xukan Ran. 2019. Deep learning with edge computing: A review. *Proc. IEEE* 107, 8 (2019), 1655–1674.

[28] Richard J Chen, Judy J Wang, Drew FK Williamson, Tiffany Y Chen, Jana Lipkova, Ming Y Lu, Sharifa Sahai, and Faisal Mahmood. 2023. Algorithmic fairness in artificial intelligence for medicine and healthcare. *Nature biomedical engineering* 7, 6 (2023), 719–742.

[29] Wei-Hao Chen, Chunmeng Dou, Kai-Xiang Li, Wei-Yu Lin, Pin-Yi Li, Jian-Hao Huang, Jing-Hong Wang, et al. 2019. CMOS-integrated memristive non-volatile computing-in-memory for AI edge processors. *Nature Electronics* 2, 9 (2019), 420–428.

[30] Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. 2024. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[31] Pau-Chen Cheng, Wojciech Ozga, Enriquillo Valdez, Salman Ahmed, Zhongshu Gu, Hani Jamjoom, Hubertus Franke, and James Bottomley. 2024. Intel tdx demystified: A top-down approach. *Comput. Surveys* 56, 9 (2024), 1–33.

[32] Vladimir Chikin and Mikhail Antiukh. 2022. Data-Free Network Compression via Parametric Non-uniform Mixed Precision Quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 450–459.

[33] Krishna Teja Chitty-Venkata and Arun K Somani. 2022. Neural architecture search survey: A hardware perspective. *Comput. Surveys* 55, 4 (2022), 1–36.

[34] Hyungmin Cho, Jeesoo Lee, and Jaejin Lee. 2021. FARNN: FPGA-GPU hybrid acceleration platform for recurrent neural networks. *IEEE Transactions on Parallel and Distributed Systems* 33, 7 (2021), 1725–1738.

[35] Ziaul Choudhury, Shashwat Shrivastava, Lavanya Ramapantulu, and Suresh Purini. 2022. An FPGA overlay for CNN inference with fine-grained flexible parallelism. *ACM Transactions on Architecture and Code Optimization (TACO)* 19, 3 (2022), 1–26.

[36] Yinghao Chu, Daquan Feng, Zuozhu Liu, Lei Zhang, Zizhou Zhao, Zhenzhong Wang, Zhiyong Feng, and Xiang-Gen Xia. 2022. A Fine-Grained Attention Model for High Accuracy Operational Robot Guidance. *IEEE Internet of Things Journal* (2022).

[37] Hanshuai Cui, Zhiqing Tang, Jiong Lou, Weijia Jia, and Wei Zhao. 2024. Latency-Aware Container Scheduling in Edge Cluster Upgrades: A Deep Reinforcement Learning Approach. *IEEE Transactions on Services Computing* (2024).

[38] Yufei Cui, Shangyu Wu, Qiao Li, Antoni B Chan, Tei-Wei Kuo, and Chun Jason Xue. 2022. Bits-Ensemble: Toward Light-Weight Robust Deep Ensemble by Bits-Sharing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 41, 11 (2022), 4397–4408.

[39] John P Cunningham and Zoubin Ghahramani. 2015. Linear dimensionality reduction: Survey, insights, and generalizations. *The Journal of Machine Learning Research* 16, 1 (2015), 2859–2900.

[40] Yueyue Dai, Ke Zhang, Sabita Maharjan, and Yan Zhang. 2020. Edge intelligence for energy-efficient computation offloading and resource allocation in 5G beyond. *IEEE Transactions on Vehicular Technology* 69, 10 (2020), 12175–12186.

[41] Hans Jakob Damsgaard, Antoine Grenier, Dewant Katare, Zain Taufique, Salar Shakibhamedan, Tiago Troccoli, Georgios Chatzitsompanis, Anil Kanduri, Aleksandr Ometov, Aaron Yi Ding, et al. 2024. Adaptive approximate computing in edge AI and IoT applications: A review. *Journal of Systems Architecture* (2024), 103114.

[42] Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie. 2020. Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proc. IEEE* 108, 4 (2020), 485–532.

[43] Shuiguang Deng, Hailiang Zhao, Weijia Fang, Jianwei Yin, Schahram Dustdar, and Albert Y Zomaya. 2020. Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal* 7, 8 (2020), 7457–7469.

[44] Sauptik Dhar, Junyao Guo, Jiayi Liu, Samarth Tripathi, Unmesh Kurup, and Mohak Shah. 2021. A survey of on-device machine learning: An algorithms and learning theory perspective. *ACM Transactions on Internet of Things* 2, 3 (2021), 1–49.

[45] Yaoyao Ding, Cody Hao Yu, et al. 2023. Hidet: Task-mapping programming paradigm for deep learning tensor programs. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*. 370–384.

[46] Thanh-Toan Do, Tuan Hoang, Victor Pomponiu, Yiren Zhou, Zhao Chen, Ngai-Man Cheung, et al. 2018. Accessible melanoma detection using smartphones and mobile image analysis. *IEEE Transactions on Multimedia* 20, 10 (2018), 2849–2864.

[47] Shi Dong, Junxiao Tang, Khushnood Abbas, Ruizhe Hou, Joarder Kamruzzaman, Leszek Rutkowski, and Rajkumar Buyya. 2024. Task offloading strategies for mobile edge computing: A survey. *Computer Networks* (2024), 110791.

[48] Jun Du, Tianyi Lin, Chunxiao Jiang, Qianqian Yang, C Faouzi Bader, and Zhu Han. 2024. Distributed Foundation Models for Multi-Modal Learning in 6G Wireless Networks. *IEEE Wireless Communications* 31, 3 (2024), 20–30.

[49] Sijing Duan, Dan Wang, Ju Ren, Feng Lyu, Ye Zhang, Huaqing Wu, and Xuemin Shen. 2022. Distributed artificial intelligence empowered by end-edge-cloud computing: A survey. *IEEE Communications Surveys & Tutorials* 25, 1 (2022), 591–624.

[50] Extrapolate. 2024. 10 Best Edge Computing Devices For Edge AI Applications. *Extrapolate* (2024). https://www.extrapolate.com/blog/top-10-edge-computing-devices-2024

[51] Huan Feng, Kassem Fawaz, and Kang G Shin. 2017. Continuous authentication for voice assistants. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. 343–355.

[52] Jeremy Fowers, Kalin Ovtcharov, Michael Papamichael, Todd Massengill, Ming Liu, Daniel Lo, Shlomi Alkalay, Michael Haselman, Logan Adams, Mahdi Ghandi, et al. 2018. A configurable cloud-scale DNN processor for real-time AI. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 1–14.

[53] Morgan R Frank, David Autor, James E Bessen, Erik Brynjolfsson, Manuel Cebrian, David J Deming, Maryann Feldman, Matthew Groh, José Lobo, Esteban Moro, et al. 2019. Toward understanding the impact of artificial intelligence on labor. *Proceedings of the National Academy of Sciences* 116, 14 (2019), 6531–6539.

[54] Yonggan Fu, Haoran You, Yang Zhao, Yue Wang, Chaojian Li, et al. 2020. Fractrain: Fractionally squeezing bit savings both temporally and spatially for efficient dnn training. *Advances in Neural Information Processing Systems* 33 (2020), 12127–12139.

[55] Chang Gao, Antonio Rios-Navarro, Xi Chen, Shih-Chii Liu, and Tobi Delbruck. 2020. EdgeDRNN: Recurrent neural network accelerator for edge inference. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 10, 4 (2020), 419–432.

[56] Shangqian Gao, Feihu Huang, Jian Pei, and Heng Huang. 2020. Discrete model compression with resource constraint for deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1899–1908.

[57] Pedro Garcia Lopez, Alberto Montresor, Dick Epema, Anwitaman Datta, Teruo Higashino, Adriana Iamnitchi, Marinho Barcellos, Pascal Felber, and Etienne Riviere. 2015. Edge-centric computing: Vision and challenges. 37–42 pages.

[58] Tong Geng, Ang Li, Tianqi Wang, Chunshu Wu, et al. 2020. O3BNN-R: An out-of-order architecture for high-performance and regularized BNN inference. *IEEE Transactions on Parallel and Distributed Systems* 32, 1 (2020), 199–213.

[59] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2022. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*. Chapman and Hall/CRC, 291–326.

[60] Bob Gill and Santhosh Rao. 2017. *Technology Insight: Edge Computing in Support of the Internet of Things*. Technical Report. Gartner Research Report.

[61] Fateneh Golpayegani, Nanxi Chen, Nima Afraz, Eric Gyamfi, Abdollah Malekjafarian, Dominik Schäfer, and Christian Krupitzer. 2024. Adaptation in edge computing: a review on design principles and research challenges. *ACM Transactions on Autonomous and Adaptive Systems* 19, 3 (2024), 1–43.

[62] Taiyuan Gong, Li Zhu, F Richard Yu, and Tao Tang. 2023. Edge intelligence in intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems* 24, 9 (2023), 8919–8944.

[63] Huixian Gu, Liqiang Zhao, Zhu Han, Gan Zheng, and Shenghui Song. 2023. AI-Enhanced Cloud-Edge-Terminal Collaborative Network: Survey, Applications, and Future Directions. *IEEE Communications Surveys & Tutorials* (2023).

[64] Lin Gu, Deze Zeng, Jie Hu, Hai Jin, Song Guo, and Albert Y Zomaya. 2021. Exploring layered container structure for cost efficient microservice deployment. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 1–9.

[65] Miguel A Guillén, Antonio Llanes, Baldomero Imbernón, Raquel Martínez-España, Andrés Bueno-Crespo, Juan-Carlos Cano, and José M Cecilia. 2021. Performance evaluation of edge-computing platforms for the prediction of low temperatures in agriculture using deep learning. *The Journal of Supercomputing* 77 (2021), 818–840.

[66] Jialin Guo, Jie Wu, Anfeng Liu, and Neal N Xiong. 2021. LightFed: An efficient and secure federated edge learning system on model splitting. *IEEE Transactions on Parallel and Distributed Systems* 33, 11 (2021), 2701–2713.

[67] Yundi Guo, Beiji Zou, Ju Ren, Qingqing Liu, Deyu Zhang, and Yaoxue Zhang. 2019. Distributed and efficient object detection via interactions among devices, edge, and cloud. *IEEE Transactions on Multimedia* 21, 11 (2019), 2903–2915.

[68] Chirag Gupta, Arun Sai Suggala, Ankit Goyal, Harsha Vardhan Simhadri, Bhargavi Paranjape, Ashish Kumar, Saurabh Goyal, Raghavendra Udupa, Manik Varma, and Prateek Jain. 2017. Protonn: Compressed and accurate knn for resource-scarce devices. In *International Conference on Machine Learning*. PMLR, 1331–1340.

[69] MyungJoo Ham, Jijoong Moon, Geunsik Lim, et al. 2021. NNStreamer: Efficient and Agile Development of On-Device AI Systems. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE, 198–207.

[70] MyungJoo Ham, Sangjung Woo, Jaeyun Jung, Wook Song, et al. 2022. Toward among-device AI from on-device AI with stream pipelines. In *Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice*. 285–294.

[71] Song Han, Huizi Mao, and William J Dally. 2016. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *International Conference on Learning Representations (ICLR)* (2016).

[72] Tao Han, Khan Muhammad, Tanveer Hussain, Jaime Lloret, and Sung Wook Baik. 2020. An efficient deep learning framework for intelligent energy management in IoT networks. *IEEE Internet of Things Journal* 8, 5 (2020), 3170–3179.

[73] Brooks Hanson, Shelley Stall, Joel Cutcher-Gershenfeld, Kristina Vrouwenvelder, Christopher Wirz, Yuhan Rao, and Ge Peng. 2023. Garbage in, garbage out: mitigating risks and maximizing benefits of AI in research. *Nature* 623, 7985 (2023), 28–31.

[74] Yixue Hao, Yiming Miao, Long Hu, M Shamim Hossain, Ghulam Muhammad, and Syed Umar Amin. 2019. Smart-Edge-CoCaCo: AI-enabled smart edge with joint computation, caching, and communication in heterogeneous IoT. *IEEE Network* 33, 2 (2019), 58–64.

[75] Zhiwei Hao, Yong Luo, Zhi Wang, Han Hu, and Jianping An. 2022. CDFKD-MFS: Collaborative Data-Free Knowledge Distillation via Multi-Level Feature Sharing. *IEEE Transactions on Multimedia* 24 (2022), 4262–4274.

[76] Vahideh Hayyolalam, Moayad Aloqaily, Öznur Özkasap, and Mohsen Guizani. 2021. Edge intelligence for empowering IoT-based healthcare systems. *IEEE Wireless Communications* 28, 3 (2021), 6–14.

[77] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[78] Fred Hohman, Mary Beth Kery, Donghao Ren, and Dominik Moritz. 2024. Model compression in practice: Lessons learned from practitioners creating on-device machine learning experiences. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–18.

[79] Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dynabert: Dynamic bert with adaptive width and depth. *Advances in Neural Information Processing Systems* 33 (2020), 9782–9793.

[80] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, et al. 2019. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1314–1324.

[81] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[82] Haochen Hua, Yutong Li, Tonghe Wang, Nanqing Dong, Wei Li, and Junwei Cao. 2023. Edge Computing with Artificial Intelligence: A Machine Learning Perspective. *Comput. Surveys* 55, 9 (2023), 1–35.
[83] Anbu Huang, Yang Liu, Tianjian Chen, Yongkai Zhou, Quan Sun, Hongfeng Chai, and Qiang Yang. 2021. Starfl: Hybrid federated learning architecture for smart urban computing. *ACM Transactions on Intelligent Systems and Technology (TIST)* 12, 4 (2021), 1–23.
[84] Chun-Hsian Huang, Wen-Tung Chen, Yi-Chun Chang, and Kuan-Ting Wu. 2024. An Edge and Trustworthy AI UAV System With Self-Adaptivity and Hyperspectral Imaging for Air Quality Monitoring. *IEEE Internet of Things Journal* (2024).
[85] Kai Huang and Wei Gao. 2022. Real-time neural network inference on extremely weak devices: agile offloading with explainable AI. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*. 200–213.
[86] Yakun Huang, Xiuquan Qiao, Jian Tang, Pei Ren, et al. 2020. Deepadapter: A collaborative deep learning framework for the mobile web using context-aware network pruning. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 834–843.
[87] Yakun Huang, Xiuquan Qiao, Jian Tang, Pei Ren, Ling Liu, Calton Pu, and Junliang Chen. 2021. An integrated cloud-edge-device adaptive deep learning service for cross-platform web. *IEEE Transactions on Mobile Computing* 22, 4 (2021), 1950–1967.
[88] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360* (2016).
[89] Intel. 2018. $OpenVINO^{TM}$ Toolkit repository. https://github.com/openvinotoolkit/openvino. *GitHub repository* (2018).
[90] Nitthilan Kannappan Jayakodi, Janardhan Rao Doppa, and Partha Pratim Pande. 2021. A general hardware and software co-design framework for energy-efficient edge AI. In *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE, 1–7.
[91] Tianyu Jia, Yuhao Ju, et al. 2020. Ncpu: An embedded neural cpu architecture on resource-constrained low power devices for real-time end-to-end performance. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 1097–1109.
[92] Hai Jin, Dongshan Bai, Dezhong Yao, Yutong Dai, Lin Gu, Chen Yu, and Lichao Sun. 2022. Personalized edge intelligence via federated self-knowledge distillation. *IEEE Transactions on Parallel and Distributed Systems* 34, 2 (2022), 567–580.
[93] Keren J Kanarik, Wojciech T Osowiecki, Yu Lu, Dipongkar Talukder, Niklas Roschewsky, Sae Na Park, Mattan Kamon, David M Fried, and Richard A Gottscho. 2023. Human–machine collaboration for improving semiconductor process development. *Nature* 616, 7958 (2023), 707–711.
[94] Dewant Katare, Diego Perino, Jari Nurmi, Martijn Warnier, Marijn Janssen, and Aaron Yi Ding. 2023. A survey on approximate edge AI for energy efficient autonomous driving services. *IEEE Communications Surveys & Tutorials* (2023).
[95] Ibrahim Kök, Feyza Yıldırım Okay, Özgecan Muyanlı, and Suat Özdemir. 2023. Explainable artificial intelligence (xai) for internet of things: a survey. *IEEE Internet of Things Journal* 10, 16 (2023), 14764–14779.
[96] Linghe Kong, Jinlin Tan, Junqin Huang, Guihai Chen, Shuaitian Wang, Xi Jin, Peng Zeng, Muhammad Khan, and Sajal K Das. 2022. Edge-computing-driven internet of things: A survey. *Comput. Surveys* 55, 8 (2022), 1–41.
[97] Alexandros Kouris, Stylianos I Venieris, Stefanos Laskaridis, and Nicholas D Lane. 2022. Fluid Batching: Exit-Aware Preemptive Serving of Early-Exit Neural Networks on Edge NPUs. *arXiv preprint arXiv:2209.13443* (2022).
[98] Bo-Han Kung, Po-Yuan Hu, Chiu-Chang Huang, Cheng-Che Lee, Chia-Yu Yao, and Chieh-Hsiung Kuan. 2020. An efficient ECG classification system using resource-saving architecture and random forest. *IEEE Journal of Biomedical and Health Informatics* 25, 6 (2020), 1904–1914.
[99] Matt Kusner, Stephen Tyree, Kilian Weinberger, and Kunal Agrawal. 2014. Stochastic neighbor compression. In *International conference on machine learning*. PMLR, 622–630.
[100] Woosuk Kwon, Sehoon Kim, Michael W Mahoney, Joseph Hassoun, Kurt Keutzer, and Amir Gholami. 2022. A Fast Post-Training Pruning Framework for Transformers. (2022).
[101] Khaled B Letaief, Yuanming Shi, Jianmin Lu, and Jianhua Lu. 2021. Edge artificial intelligence for 6G: Vision, enabling technologies, and applications. *IEEE Journal on Selected Areas in Communications* 40, 1 (2021), 5–36.
[102] Bai Li, Yakun Ouyang, Li Li, and Youmin Zhang. 2022. Autonomous driving on curvy roads without reliance on frenet frame: A cartesian-based trajectory planning method. *IEEE Transactions on Intelligent Transportation Systems* 23, 9 (2022), 15729–15741.
[103] Bing Li, Songyun Qu, and Ying Wang. 2021. An automated quantization framework for high-utilization rram-based pim. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 41, 3 (2021), 583–596.
[104] Baoting Li, Danqing Zhang, Pengfei Zhao, Hang Wang, Xuchong Zhang, Hongbin Sun, and Nanning Zheng. 2024. DQ-STP: An Efficient Sparse On-Device Training Processor Based on Low-Rank Decomposition and Quantization for DNN. *IEEE Transactions on Circuits and Systems I: Regular Papers* 71, 4 (2024), 1665–1678.
[105] En Li, Liekang Zeng, Zhi Zhou, and Xu Chen. 2019. Edge AI: On-demand accelerating deep neural network inference via edge computing. *IEEE Transactions on Wireless Communications* 19, 1 (2019), 447–457.
[106] Guangli Li, Xiu Ma, Xueying Wang, Lei Liu, et al. 2020. Fusion-catalyzed pruning for optimizing deep learning on intelligent edge devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, 11 (2020), 3614–3626.
[107] Liangzhi Li, Kaoru Ota, and Mianxiong Dong. 2018. Deep learning for smart industry: Efficient manufacture inspection system with fog computing. *IEEE Transactions on Industrial Informatics* 14, 10 (2018), 4665–4673.

[108] Qiushi Li, Ju Ren, Xinglin Pan, et al. 2022. ENIGMA: Low-Latency and Privacy-Preserving Edge Inference on Heterogeneous Neural Network Accelerators. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 458–469.
[109] Tao Li, Yitao Ma, and Tetsuo Endoh. 2022. From algorithm to module: Adaptive and energy-efficient quantization method for edge artificial intelligence in iot society. *IEEE Transactions on Industrial Informatics* 19, 8 (2022), 8953–8964.
[110] Tan Li, Hong Wang, Dongxu Pan, Jiasheng Tan, Junxu Hou, Lingjie Kong, and Jingbo Liu. 2024. A machine vision approach with temporal fusion strategy for concrete vibration quality monitoring. *Applied Soft Computing* 160 (2024), 111684.
[111] Xian Li and Suzhi Bi. 2024. Optimal AI Model Splitting and Resource Allocation for Device-Edge Co-Inference in Multi-User Wireless Sensing Systems. *IEEE Transactions on Wireless Communications* (2024).
[112] Xiong Li, Shanpeng Liu, Fan Wu, Saru Kumari, and Joel JPC Rodrigues. 2018. Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications. *IEEE Internet of Things Journal* 6, 3 (2018), 4755–4763.
[113] Yunsheng Li, Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Lu Yuan, Zicheng Liu, Lei Zhang, and Nuno Vasconcelos. 2020. MicroNet: Towards image recognition with extremely low FLOPs. *arXiv preprint arXiv:2011.12289* (2020).
[114] Qianlin Liang, Walid A Hanafy, Ahmed Ali-Eldin, and Prashant Shenoy. 2023. Model-driven cluster resource management for ai workloads in edge clouds. *ACM Transactions on Autonomous and Adaptive Systems* 18, 1 (2023), 1–26.
[115] Siyuan Liang, Hao Wu, Li Zhen, et al. 2022. Edge YOLO: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems* 23, 12 (2022), 25345–25360.
[116] Wei Yang Bryan Lim, Jer Shyuan Ng, Zehui Xiong, Jiangming Jin, et al. 2021. Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning. *IEEE Transactions on Parallel and Distributed Systems* 33, 3 (2021), 536–550.
[117] Chuan Lin, Guangjie Han, Jinfang Jiang, Chao Li, Syed Bilal Hussain Shah, and Qian Liu. 2023. Underwater pollution tracking based on software-defined multi-tier edge computing in 6G-based underwater wireless networks. *IEEE Journal on Selected Areas in Communications* 41, 2 (2023), 491–503.
[118] Mingbao Lin, Rongrong Ji, Yan Wang, Yichen Zhang, Baochang Zhang, Yonghong Tian, and Ling Shao. 2020. Hrank: Filter pruning using high-rank feature map. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1529–1538.
[119] Zehong Lin, Suzhi Bi, and Ying-Jun Angela Zhang. 2021. Optimizing AI service placement and resource allocation in mobile edge intelligence systems. *IEEE Transactions on Wireless Communications* 20, 11 (2021), 7257–7271.
[120] Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2018. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055* (2018).
[121] Peng Liu, Bozhao Qi, and Suman Banerjee. 2018. Edgeeye: An edge service framework for real-time intelligent video analytics. In *Proceedings of the 1st International Workshop on Edge Systems, Analytics and Networking*. 1–6.
[122] Shaoshan Liu, Liangkai Liu, Jie Tang, Bo Yu, Yifan Wang, and Weisong Shi. 2019. Edge computing for autonomous driving: Opportunities and challenges. *Proc. IEEE* 107, 8 (2019), 1697–1716.
[123] Xiaochen Liu, Yurong Jiang, Puneet Jain, and Kyu-Han Kim. 2018. TAR: Enabling fine-grained targeted advertising in retail stores. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. 323–336.
[124] Yang Liu, Zhuo Ma, Ximeng Liu, Siqi Ma, and Kui Ren. 2019. Privacy-preserving object detection for medical images with faster R-CNN. *IEEE Transactions on Information Forensics and Security* 17 (2019), 69–84.
[125] Yaqiong Liu, Mugen Peng, Guochu Shou, Yudong Chen, and Siyu Chen. 2020. Toward edge intelligence: Multiaccess edge computing for 5G and Internet of Things. *IEEE Internet of Things Journal* 7, 8 (2020), 6722–6747.
[126] Yinghan Long, Indranil Chakraborty, Gopalakrishnan Srinivasan, and Kaushik Roy. 2021. Complexity-aware adaptive training and inference for edge-cloud distributed AI systems. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 573–583.
[127] Jiong Lou, Hao Luo, Zhiqing Tang, Weijia Jia, and Wei Zhao. 2022. Efficient Container Assignment and Layer Sequencing in Edge Computing. *IEEE Transactions on Services Computing* (2022).
[128] Jiong Lou, Zhiqing Tang, and Weijia Jia. 2022. Energy-efficient Joint Task Assignment and Migration in Data Centers: A Deep Reinforcement Learning Approach. *IEEE Transactions on Network and Service Management* (2022).
[129] Jiong Lou, Zhiqing Tang, Weijia Jia, Wei Zhao, and Jie Li. 2023. Startup-aware Dependent Task Scheduling with Bandwidth Constraints in Edge Computing. *IEEE Transactions on Mobile Computing* (2023).
[130] Jiong Lou, Zhiqing Tang, Songli Zhang, Weijia Jia, Wei Zhao, and Jie Li. 2022. Cost-Effective Scheduling for Dependent Tasks With Tight Deadline Constraints in Mobile Edge Computing. *IEEE Transactions on Mobile Computing* (2022).
[131] Haodong Lu, Miao Du, Xiaoming He, Kai Qian, Jianli Chen, Yanfei Sun, and Kun Wang. 2021. An adaptive neural architecture search design for collaborative edge-cloud computing. *IEEE Network* 35, 5 (2021), 83–89.
[132] Chunjie Luo, Xiwen He, Jianfeng Zhan, Lei Wang, Wanling Gao, and Jiahui Dai. 2020. Comparison and benchmarking of ai models and frameworks on mobile devices. *arXiv preprint arXiv:2005.05085* (2020).
[133] Yandong Luo and Shimeng Yu. 2021. AILC: Accelerate on-chip incremental learning with compute-in-memory technology. *IEEE Trans. Comput.* 70, 8 (2021), 1225–1238.
[134] Wenkai Lv, Quan Wang, Pengfei Yang, Yunqing Ding, Bijie Yi, Zhenyi Wang, and Chengmin Lin. 2022. Microservice deployment in edge computing based on deep q learning. *IEEE Transactions on Parallel and Distributed Systems* 33, 11 (2022), 2968–2978.

[135] Zhihan Lv, Liang Qiao, and Sahil Verma. 2021. AI-enabled IoT-edge data analytics for connected living. *ACM Transactions on Internet Technology* 21, 4 (2021), 1–20.
[136] Bo Lyu, Shiping Wen, Kaibo Shi, and Tingwen Huang. 2021. Multiobjective reinforcement learning-based neural architecture search for efficient portrait parsing. *IEEE Transactions on Cybernetics* (2021).
[137] Bo Lyu, Hang Yuan, Longfei Lu, and Yunye Zhang. 2021. Resource-constrained neural architecture search on edge devices. *IEEE Transactions on Network Science and Engineering* 9, 1 (2021), 134–142.
[138] Huirong Ma, Zhi Zhou, Xiaoxi Zhang, and Xu Chen. 2023. Toward carbon-neutral edge computing: Greening edge AI by harnessing spot and future carbon markets. *IEEE Internet of Things Journal* 10, 18 (2023), 16637–16649.
[139] Lichuan Ma, Qingqi Pei, Lu Zhou, Haojin Zhu, Licheng Wang, and Yusheng Ji. 2020. Federated data cleaning: Collaborative and privacy-preserving data cleaning for edge intelligence. *IEEE Internet of Things Journal* 8, 8 (2020), 6757–6770.
[140] Lele Ma, Shanhe Yi, Nancy Carter, and Qun Li. 2018. Efficient live migration of edge services leveraging container layered storage. *IEEE Transactions on Mobile Computing* 18, 9 (2018), 2020–2033.
[141] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. 2018. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 116–131.
[142] Yuyi Mao, Xianghao Yu, Kaibin Huang, Ying-Jun Angela Zhang, and Jun Zhang. 2024. Green edge AI: A contemporary survey. *Proc. IEEE* (2024).
[143] Alberto Marchisio, Beatrice Bussolino, Alessio Colucci, Maurizio Martina, Guido Masera, and Muhammad Shafique. 2020. Q-capsnets: A specialized framework for quantizing capsule networks. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 1–6.
[144] Sven Mayer, Gierad Laput, and Chris Harrison. 2020. Enhancing mobile voice assistants with worldgaze. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–10.
[145] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)* 54, 6 (2021), 1–35.
[146] Sachin Mehta and Mohammad Rastegari. 2021. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. (2021).
[147] Alexander Menshchikov, Dmitrii Shadrin, Viktor Prutyanov, Daniil Lopatkin, Sergey Sosnin, et al. 2021. Real-time detection of hogweed: UAV platform empowered by deep learning. *IEEE Trans. Comput.* 70, 8 (2021), 1175–1188.
[148] Christian Meurisch and Max Mühlhäuser. 2021. Data protection in AI services: A survey. *ACM Computing Surveys (CSUR)* 54, 2 (2021), 1–38.
[149] Fengchun Miao, Wayne Holmes, Ronghuai Huang, Hui Zhang, et al. 2021. *AI and education: A guidance for policymakers*. Unesco Publishing.
[150] Qinghai Miao, Wenbo Zheng, Yisheng Lv, Min Huang, Wenwen Ding, and Fei-Yue Wang. 2023. DAO to HANOI via DeSci: AI paradigm shifts from AlphaGo to ChatGPT. *IEEE/CAA Journal of Automatica Sinica* 10, 4 (2023), 877–897.
[151] Microsoft. 2019. ONNX Runtime: cross-platform, high performance ML inferencing and training accelerator. https://github.com/microsoft/onnxruntime. *GitHub repository* (2019).
[152] Rahul Mishra, Ashish Gupta, and Hari Prabat Gupta. 2021. Locomotion mode recognition using sensory data with noisy labels: A deep learning approach. *IEEE Transactions on Mobile Computing* (2021).
[153] Sparsh Mittal. 2019. A survey on optimized implementation of deep learning models on the nvidia jetson platform. *Journal of Systems Architecture* 97 (2019), 428–442.
[154] Marissa Mock, Suzanne Edavettal, Christopher Langmead, and Alan Russell. 2023. AI can help to speed up drug discovery—but only if we give it the right data. *Nature* 621, 7979 (2023), 467–470.
[155] Alejandro Moran, Christiam F Frasser, Miquel Roca, and Josep L Rossello. 2019. Energy-efficient pattern recognition hardware with elementary cellular automata. *IEEE Trans. Comput.* 69, 3 (2019), 392–401.
[156] Chamin Morikawa, Michihiro Kobayashi, Masaki Satoh, Yasuhiro Kuroda, Teppei Inomata, Hitoshi Matsuo, Takeshi Miura, and Masaki Hilaga. 2021. Image and video processing on mobile devices: a survey. *The Visual Computer* 37, 12 (2021), 2931–2949.
[157] Fangyi Mou, Jiong Lou, Zhiqing Tang, Yuan Wu, Weijia Jia, Yan Zhang, and Wei Zhao. 2024. Adaptive Digital Twin Migration in Vehicular Edge Computing and Networks. *IEEE Transactions on Vehicular Technology* (2024).
[158] MG Sarwar Murshed, Christopher Murphy, Daqing Hou, Nazar Khan, Ganesh Ananthanarayanan, and Faraz Hussain. 2021. Machine learning at the network edge: A survey. *ACM Computing Surveys (CSUR)* 54, 8 (2021), 1–37.
[159] Yasser Nabil, Hesham ElSawy, Suhail Al-Dharrab, Hassan Mostafa, and Hussein Attia. 2022. Data aggregation in regular large-scale IoT networks: Granularity, reliability, and delay tradeoffs. *IEEE Internet of Things Journal* 9, 18 (2022), 17767–17784.
[160] Jianyuan Ni, Raunak Sarbajna, Yang Liu, et al. 2022. Cross-modal knowledge distillation for vision-to-sensor action recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4448–4452.
[161] Xuefei Ning, Guangjun Ge, Wenshuo Li, Zhenhua Zhu, Yin Zheng, Xiaoming Chen, et al. 2021. FTT-NAS: Discovering fault-tolerant convolutional neural architecture. *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 26, 6 (2021), 1–24.

[162] Zhaolong Ning, Kaiyuan Zhang, Xiaojie Wang, Lei Guo, et al. 2020. Intelligent edge computing in internet of vehicles: a joint computation offloading and caching solution. *IEEE Transactions on Intelligent Transportation Systems* 22, 4 (2020), 2212–2225.
[163] Wei Niu, Xiaolong Ma, Sheng Lin, Shihao Wang, Xuehai Qian, Xue Lin, Yanzhi Wang, and Bin Ren. 2020. Patdnn: Achieving real-time dnn execution on mobile devices with pattern-based weight pruning. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*. 907–922.
[164] Anant V Nori, Rahul Bera, et al. 2021. Reduct: Keep it close, keep it cool!: Efficient scaling of dnn inference on multi-core cpus with near-cache compute. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 167–180.
[165] Jose Nunez-Yanez and Neil Howard. 2021. Energy-efficient neural networks with near-threshold processors and hardware accelerators. *Journal of Systems Architecture* 116 (2021), 102062.
[166] NVIDIA. 2017. $NVIDIA^{®}\ TensorRT^{TM}$, an SDK for high-performance deep learning inference. https://github.com/NVIDIA/TensorRT. *GitHub repository* (2017).
[167] Anton Obukhov, Maxim Rakhuba, Stamatios Georgoulis, Menelaos Kanakis, Dengxin Dai, and Luc Van Gool. 2020. T-basis: a compact representation for neural networks. In *International Conference on Machine Learning*. PMLR, 7392–7404.
[168] Francesco Paissan, Alberto Ancilotto, and Elisabetta Farella. 2022. PhiNets: a scalable backbone for low-power AI at the edge. *ACM Transactions on Embedded Computing Systems* 21, 5 (2022), 1–18.
[169] Jihong Park, Sumudu Samarakoon, Mehdi Bennis, and Mérouane Debbah. 2019. Wireless network intelligence at the edge. *Proc. IEEE* 107, 11 (2019), 2204–2239.
[170] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).
[171] Biagio Peccerillo, Mirco Mannino, Andrea Mondelli, and Sandro Bartolini. 2022. A survey on hardware accelerators: Taxonomy, trends, challenges, and perspectives. *Journal of Systems Architecture* 129 (2022), 102561.
[172] Giacomo Pedretti, Catherine E Graves, Sergey Serebryakov, Ruibin Mao, Xia Sheng, Martin Foltin, Can Li, and John Paul Strachan. 2021. Tree-based machine learning performed in-memory with memristive analog CAM. *Nature communications* 12, 1 (2021), 5806.
[173] István Pelle, János Czentye, János Dóka, and Balázs Sonkoly. 2020. Dynamic latency control of serverless applications operated on aws lambda and greengrass. In *Proceedings of the SIGCOMM'20 Poster and Demo Sessions*. 33–34.
[174] Flavio Ponzina, Marco Rios, Alexandre Levisse, Giovanni Ansaloni, and David Atienza. 2023. Overflow-free compute memories for edge AI acceleration. *ACM Transactions on Embedded Computing Systems* 22, 5s (2023), 1–23.
[175] Rachmad Vidya Wicaksana Putra et al. 2020. Fspinn: An optimization framework for memory-efficient and energy-efficient spiking neural networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, 11 (2020), 3601–3613.
[176] Aji Gautama Putrada, Maman Abdurohman, Doan Perdana, and Hilal Hudan Nuha. 2023. Q8KNN: A Novel 8-Bit KNN Quantization Method for Edge Computing in Smart Lighting Systems with NodeMCU. In *Intelligent Systems Conference*. Springer, 598–615.
[177] Tie Qiu, Jiancheng Chi, Xiaobo Zhou, Zhaolong Ning, Mohammed Atiquzzaman, and Dapeng Oliver Wu. 2020. Edge computing in industrial internet of things: Architecture, advances and challenges. *IEEE Communications Surveys & Tutorials* 22, 4 (2020), 2462–2488.
[178] Md Abdur Rahman and M Shamim Hossain. 2021. An internet-of-medical-things-enabled edge computing framework for tackling COVID-19. *IEEE Internet of Things Journal* 8, 21 (2021), 15847–15854.
[179] Mohammad Saidur Rahman, Ibrahim Khalil, et al. 2020. Towards privacy preserving AI based composition framework in edge networks using fully homomorphic encryption. *Engineering Applications of Artificial Intelligence* 94 (2020), 103737.
[180] Qiqi Ren, Omid Abbasi, Gunes Karabulut Kurt, et al. 2022. Caching and computation offloading in high altitude platform station (HAPS) assisted intelligent transportation systems. *IEEE Transactions on Wireless Communications* 21, 11 (2022), 9010–9024.
[181] Arman Roohi, Shadi Sheikhfaal, Shaahin Angizi, Deliang Fan, and Ronald F DeMara. 2019. Apgan: Approximate gan for robust low energy learning from imprecise components. *IEEE Trans. Comput.* 69, 3 (2019), 349–360.
[182] Drew Roselli, Jeanna Matthews, and Nisha Talagala. 2019. Managing bias in AI. In *Companion proceedings of the 2019 world wide web conference*. 539–544.
[183] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4510–4520.
[184] Iqbal H Sarker, Mohammed Moshiul Hoque, Md Kafil Uddin, and Tawfeeq Alsanoosy. 2021. Mobile data science and intelligent apps: concepts, AI-based modeling and research directions. *Mobile Networks and Applications* 26, 1 (2021), 285–303.
[185] Mahadev Satyanarayanan, Paramvir Bahl, Ramón Caceres, and Nigel Davies. 2009. The case for vm-based cloudlets in mobile computing. *IEEE pervasive Computing* 8, 4 (2009), 14–23.
[186] Jiawei Shao and Jun Zhang. 2020. Bottlenet++: An end-to-end approach for feature compression in device-edge co-inference systems. In *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 1–6.
[187] Yi Sheng, Junhuan Yang, Yawen Wu, Kevin Mao, Yiyu Shi, Jingtong Hu, Weiwen Jiang, and Lei Yang. 2022. The larger the fairer? small neural networks can achieve fairness for edge devices. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*. 163–168.

[188] Yuanming Shi, Kai Yang, Tao Jiang, Jun Zhang, and Khaled B Letaief. 2020. Communication-efficient edge AI: Algorithms and systems. *IEEE Communications Surveys & Tutorials* 22, 4 (2020), 2167–2191.
[189] You Shi, Changyan Yi, Bing Chen, Chenze Yang, Kun Zhu, and Jun Cai. 2022. Joint online optimization of data sampling rate and preprocessing mode for edge–cloud collaboration-enabled industrial IoT. *IEEE Internet of Things Journal* 9, 17 (2022), 16402–16417.
[190] Md Maruf Hossain Shuvo, Syed Kamrul Islam, Jianlin Cheng, and Bashir I Morshed. 2022. Efficient acceleration of deep learning inference on resource-constrained edge devices: A review. *Proc. IEEE* 111, 1 (2022), 42–91.
[191] Vikas Sindhwani, Tara Sainath, and Sanjiv Kumar. 2015. Structured transforms for small-footprint deep learning. *Advances in Neural Information Processing Systems* 28 (2015).
[192] Soumik Sinha, Sayandeep Saha, Manaar Alam, et al. 2022. Exploring Bitslicing Architectures for Enabling FHE-Assisted Machine Learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 41, 11 (2022), 4004–4015.
[193] Tuomo Sipola, Janne Alatalo, Tero Kokkonen, and Mika Rantonen. 2022. Artificial intelligence in the IoT era: A review of edge AI hardware and software. In *2022 31st Conference of Open Innovations Association (FRUCT)*. IEEE, 320–331.
[194] Yushan Siriwardhana, Pawani Porambage, Madhusanka Liyanage, and Mika Ylianttila. 2021. A survey on mobile augmented reality with 5G mobile edge computing: Architectures, applications, and technical aspects. *IEEE Communications Surveys & Tutorials* 23, 2 (2021), 1160–1192.
[195] Ali Hassan Sodhro, Sandeep Pirbhulal, and Victor Hugo C De Albuquerque. 2019. Artificial intelligence-driven mechanism for edge computing-based industrial applications. *IEEE Transactions on Industrial Informatics* 15, 7 (2019), 4235–4243.
[196] Mengkai Song, Zhibo Wang, Zhifei Zhang, Yang Song, Qian Wang, Ju Ren, and Hairong Qi. 2020. Analyzing user-level privacy attack against federated learning. *IEEE Journal on Selected Areas in Communications* 38, 10 (2020), 2430–2444.
[197] Ayush Srivastava, Oshin Dutta, Jigyasa Gupta, et al. 2021. A variational information bottleneck based method to compress sequential networks for human action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2745–2754.
[198] Yang Sui, Miao Yin, Yu Gong, and Bo Yuan. 2024. Co-exploring structured sparsification and low-rank tensor decomposition for compact dnns. *IEEE Transactions on Neural Networks and Learning Systems* (2024).
[199] Douglas H Summerville, Kenneth M Zach, and Yu Chen. 2015. Ultra-lightweight deep packet anomaly detection for Internet of Things devices. In *2015 IEEE 34th international performance computing and communications conference (IPCCC)*. IEEE, 1–8.
[200] Danfeng Sun, Jia Wu, Jian Yang, and Huifeng Wu. 2021. Intelligent data collaboration in heterogeneous-device iot platforms. *ACM Transactions on Sensor Networks (TOSN)* 17, 3 (2021), 1–17.
[201] Danfeng Sun, Shan Xue, Huifeng Wu, and Jia Wu. 2021. A data stream cleaning system using edge intelligence for smart city industrial environments. *IEEE Transactions on Industrial Informatics* 18, 2 (2021), 1165–1174.
[202] Yang Sun, Fajie Yuan, Min Yang, Guoao Wei, et al. 2020. A generic network compression framework for sequential recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1299–1308.
[203] Tarik Taleb, Sunny Dutta, Adlen Ksentini, Muddesar Iqbal, and Hannu Flinck. 2017. Mobile edge computing potential in making cities smarter. *IEEE Communications Magazine* 55, 3 (2017), 38–43.
[204] Thierry Tambe, Coleman Hooper, Lillian Pentecost, et al. 2021. Edgebert: Sentence-level energy optimizations for latency-aware multi-task nlp inference. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*. 830–844.
[205] Thierry Tambe, En-Yu Yang, Glenn G Ko, et al. 2022. A 16-nm SoC for Noise-Robust Speech and NLP Edge AI Inference With Bayesian Sound Source Separation and Attention-Based DNNs. *IEEE Journal of Solid-State Circuits* (2022).
[206] Chong Min John Tan and Mehul Motani. 2020. Dropnet: Reducing neural network complexity via iterative pruning. In *International Conference on Machine Learning*. PMLR, 9356–9366.
[207] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*. PMLR, 6105–6114.
[208] Mingxing Tan and Quoc Le. 2021. EfficientNetV2: Smaller Models and Faster Training. In *International Conference on Machine Learning*. PMLR, 10096–10106.
[209] Qinqin Tang, Renchao Xie, Fei Richard Yu, Tianjiao Chen, Ran Zhang, Tao Huang, and Yunjie Liu. 2022. Collective deep reinforcement learning for intelligence sharing in the internet of intelligence-empowered edge computing. *IEEE Transactions on Mobile Computing* 22, 11 (2022), 6327–6342.
[210] Zhiqing Tang, Weijia Jia, Xiaojie Zhou, Wenmian Yang, and Yongjian You. 2020. Representation and reinforcement learning for task scheduling in edge computing. *IEEE Transactions on Big Data* 8, 3 (2020), 795–808.
[211] Zhiqing Tang, Jiong Lou, and Weijia Jia. 2022. Layer Dependency-aware Learning Scheduling Algorithms for Containers in Mobile Edge Computing. *IEEE Transactions on Mobile Computing* (2022).
[212] Zhiqing Tang, Fangyi Mou, Jiong Lou, Weijia Jia, Yuan Wu, and Wei Zhao. 2023. Multi-user layer-aware online container migration in edge-assisted vehicular networks. *IEEE/ACM Transactions on Networking* (2023).
[213] Zhiqing Tang, Fangyi Mou, Jiong Lou, Weijia Jia, Yuan Wu, and Wei Zhao. 2024. Joint Resource Overbooking and Container Scheduling in Edge Computing. *IEEE Transactions on Mobile Computing* (2024).

[214] Zhiqing Tang, Fuming Zhang, Xiaojie Zhou, Weijia Jia, and Wei Zhao. 2022. Pricing model for dynamic resource overbooking in edge computing. *IEEE Transactions on Cloud Computing* (2022).
[215] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118* (2024).
[216] Tencent. 2017. NCNN is a high-performance neural network inference framework optimized for the mobile platform. https://github.com/Tencent/ncnn. *GitHub repository* (2017).
[217] Tuyen X Tran, Duc V Le, Guosen Yue, and Dario Pompili. 2018. Cooperative hierarchical caching and request scheduling in a cloud radio access network. *IEEE Transactions on Mobile Computing* 17, 12 (2018), 2729–2743.
[218] Shreshth Tuli, Nipam Basumatary, Sukhpal Singh Gill, et al. 2020. HealthFog: An ensemble deep learning based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in integrated IoT and fog computing environments. *Future Generation Computer Systems* 104 (2020), 187–200.
[219] Frederick Tung and Greg Mori. 2018. Clip-q: Deep network compression learning by in-parallel pruning-quantization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7873–7882.
[220] Karen Ullrich, Edward Meeds, and Max Welling. 2017. Soft weight-sharing for neural network compression. (2017).
[221] Laurens Van Der Maaten, Eric Postma, Jaap Van den Herik, et al. 2009. Dimensionality reduction: a comparative. *J Mach Learn Res* 10, 66-71 (2009), 13.
[222] Laura Verde, Nadia Brancati, Giuseppe De Pietro, Maria Frucci, and Giovanna Sannino. 2021. A deep learning approach for voice disorder detection for smart connected living environments. *ACM Transactions on Internet Technology (TOIT)* 22, 1 (2021), 1–16.
[223] Gaurav Verma, Yashi Gupta, Abid M Malik, and Barbara Chapman. 2021. Performance evaluation of deep learning compilers for edge inference. In *2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 858–865.
[224] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. 2019. Haq: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8612–8620.
[225] Ke Wang, Ship Peng Xu, Chien-Ming Chen, SK Hafizul Islam, Mohammad Mehedi Hassan, et al. 2021. A trusted consensus scheme for collaborative learning in the edge ai computing domain. *IEEE Network* 35, 1 (2021), 204–210.
[226] Rui Wang, Zhihua Wei, Haoran Duan, Shouling Ji, Yang Long, and Zhen Hong. 2022. EfficientTDNN: Efficient architecture search for speaker recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 30 (2022), 2267–2279.
[227] Siqi Wang, Gayathri Ananthanarayanan, Yifan Zeng, et al. 2019. High-throughput cnn inference on embedded arm big. little multicore processors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, 10 (2019), 2254–2267.
[228] Sheng Wang, Zhijun Ding, and Changjun Jiang. 2020. Elastic scheduling for microservice applications in clouds. *IEEE Transactions on Parallel and Distributed Systems* 32, 1 (2020), 98–115.
[229] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. 2018. When edge meets learning: Adaptive control for resource-constrained distributed machine learning. In *IEEE INFOCOM 2018-IEEE conference on computer communications*. IEEE, 63–71.
[230] Shibo Wang, Shusen Yang, and Cong Zhao. 2020. SurveilEdge: Real-time video query based on collaborative cloud-edge deep learning. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2519–2528.
[231] Tian Wang, Haoxiong Ke, Xi Zheng, Kun Wang, Arun Kumar Sangaiah, and Anfeng Liu. 2019. Big data cleaning based on mobile edge computing in industrial sensor-cloud. *IEEE Transactions on Industrial Informatics* 16, 2 (2019), 1321–1329.
[232] Yang Wang, Dazheng Deng, Leibo Liu, et al. 2022. PL-NPU: An Energy-Efficient Edge-Device DNN Training Processor With Posit-Based Logarithm-Domain Computing. *IEEE Transactions on Circuits and Systems I: Regular Papers* 69, 10 (2022), 4042–4055.
[233] Ying Wang, Huawei Li, and Xiaowei Li. 2016. Re-architecting the on-chip memory sub-system of machine-learning accelerator for embedded devices. In *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 1–6.
[234] Yang Wang, Yubin Qin, Leibo Liu, Shaojun Wei, and Shouyi Yin. 2022. SWPU: A 126.04 TFLOPS/W edge-device sparse DNN training processor with dynamic sub-structured weight pruning. *IEEE Transactions on Circuits and Systems I: Regular Papers* 69, 10 (2022), 4014–4027.
[235] Pete Warden and Daniel Situnayake. 2019. *Tinyml: Machine learning with tensorflow lite on arduino and ultra-low-power microcontrollers*. O'Reilly Media.
[236] Liangjian Wen, Xuanyang Zhang, Haoli Bai, and Zenglin Xu. 2020. Structured pruning of recurrent neural networks through neuron selection. *Neural Networks* 123 (2020), 134–141.
[237] Steven Euijong Whang, Yuji Roh, Hwanjun Song, and Jae-Gil Lee. 2023. Data collection and quality challenges in deep learning: A data-centric ai perspective. *The VLDB Journal* 32, 4 (2023), 791–813.
[238] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, et al. 2019. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10734–10742.
[239] Junru Wu, Yue Wang, Zhenyu Wu, et al. 2018. Deep k-means: Re-training and parameter sharing with harder cluster assignments for compressing deep convolutions. In *International Conference on Machine Learning*. PMLR, 5363–5372.

[240] Jianqiu Wu, Zhongyi Yu, Jianxiong Guo, Zhiqing Tang, Tian Wang, and Weijia Jia. 2024. Two-Stage Deep Energy Optimization in IRS-Assisted UAV-Based Edge Computing Systems. *IEEE Transactions on Mobile Computing* (2024).

[241] Yirui Wu, Haifeng Guo, Chinmay Chakraborty, Mohammad R Khosravi, Stefano Berretti, and Shaohua Wan. 2022. Edge computing driven low-light image dynamic enhancement for object detection. *IEEE Transactions on Network Science and Engineering* 10, 5 (2022), 3086–3098.

[242] Ming Xia, Zunkai Huang, Li Tian, Hui Wang, Victor Chang, Yongxin Zhu, and Songlin Feng. 2021. SparkNoC: An energy-efficiency FPGA-based accelerator using optimized lightweight CNN for edge computing. *Journal of Systems Architecture* 115 (2021), 101991.

[243] Xin Xia, Hongzhi Yin, Junliang Yu, et al. 2022. On-Device Next-Item Recommendation with Self-Supervised Knowledge Distillation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 546–555.

[244] Yecheng Xiang and Hyoseung Kim. 2019. Pipelined data-parallel CPU/GPU scheduling for multi-DNN real-time inference. In *2019 IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 392–405.

[245] Jinbo Xiong, Mingfeng Zhao, Md Zakirul Alam Bhuiyan, Lei Chen, and Youliang Tian. 2019. An AI-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of IoT. *IEEE Transactions on Industrial Informatics* 17, 2 (2019), 922–933.

[246] Dianlei Xu, Tong Li, Yong Li, Xiang Su, Sasu Tarkoma, Tao Jiang, Jon Crowcroft, and Pan Hui. 2020. Edge intelligence: Architectures, challenges, and applications. *arXiv preprint arXiv:2003.12172* (2020).

[247] Dianlei Xu, Tong Li, Yong Li, Xiang Su, Sasu Tarkoma, Tao Jiang, Jon Crowcroft, and Pan Hui. 2021. Edge intelligence: Empowering intelligence to the edge of network. *Proc. IEEE* 109, 11 (2021), 1778–1837.

[248] Jiajun Xu, Zhiyuan Li, Wei Chen, Qun Wang, Xin Gao, Qi Cai, and Ziyuan Ling. 2024. On-device language models: A comprehensive review. *arXiv preprint arXiv:2409.00088* (2024).

[249] Runhua Xu, Nathalie Baracaldo, and James Joshi. 2021. Privacy-preserving machine learning: Methods, challenges and directions. *arXiv preprint arXiv:2108.04417* (2021).

[250] Zirui Xu, Fuxun Yu, Zhuwei Qin, et al. 2020. Directx: Dynamic resource-aware cnn reconfiguration framework for real-time mobile applications. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 40, 2 (2020), 246–259.

[251] Bufang Yang, Lixing He, Neiwen Ling, Zhenyu Yan, Guoliang Xing, Xian Shuai, Xiaozhe Ren, and Xin Jiang. 2023. Edgefm: Leveraging foundation model for open-set learning on the edge. In *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems*. 111–124.

[252] Huanrui Yang et al. 2020. Learning low-rank deep neural networks via singular vector orthogonality regularization and singular value sparsification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 678–679.

[253] Honghong Yang, Jinming Wen, Xiaojun Wu, Li He, and Shahid Mumtaz. 2019. An efficient edge artificial intelligence multipedestrian tracking method with rank constraint. *IEEE Transactions on Industrial Informatics* 15, 7 (2019), 4178–4188.

[254] Qing Yang and Hai Li. 2020. BitSystolic: A 26.7 TOPS/W 2b˜ 8b NPU with configurable data flows for edge devices. *IEEE Transactions on Circuits and Systems I: Regular Papers* 68, 3 (2020), 1134–1145.

[255] Jiajie Yin, Zhiqing Tang, Jiong Lou, Jianxiong Guo, Hui Cai, Xiaoming Wu, Tian Wang, and Weijia Jia. 2024. QoS-Aware Energy-Efficient Multi-UAV Offloading Ratio and Trajectory Control Algorithm in Mobile Edge Computing. *IEEE Internet of Things Journal* (2024).

[256] Haoran You, Baopu Li, Shi Huihong, Yonggan Fu, and Yingyan Lin. 2022. ShiftAddNAS: Hardware-inspired search for more accurate and efficient neural networks. In *International Conference on Machine Learning*. PMLR, 25566–25580.

[257] Sixing Yu, Phuong Nguyen, Waqwoya Abebe, et al. 2022. SPATL: salient parameter aggregation and transfer learning for heterogeneous federated learning. In *2022 SC22: International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*. IEEE Computer Society, 495–508.

[258] Yunxuan Yu, Tiandong Zhao, Kun Wang, and Lei He. 2020. Light-OPU: An FPGA-based overlay processor for lightweight convolutional neural networks. In *Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. 122–132.

[259] Jinliang Yuan, Chen Yang, Dongqi Cai, Shihe Wang, Xin Yuan, Zeling Zhang, Xiang Li, Dingge Zhang, Hanzi Mei, Xianqing Jia, et al. 2024. Mobile Foundation Model as Firmware. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. 279–295.

[260] Muhammad Zawish, Nouman Ashraf, Rafay Iqbal Ansari, and Steven Davy. 2022. Energy-aware AI-driven framework for edge-computing-based IoT applications. *IEEE Internet of Things Journal* 10, 6 (2022), 5013–5023.

[261] Liekang Zeng, Shengyuan Ye, Xu Chen, and Yang Yang. 2024. Implementation of Big AI Models for Wireless Networks with Collaborative Edge Computing. *IEEE Wireless Communications* 31, 3 (2024), 50–58.

[262] Peng Zeng, Bofeng Pan, Kim-Kwang Raymond Choo, and Hong Liu. 2020. MMDA: Multidimensional and multidirectional data aggregation for edge computing-enhanced IoT. *Journal of Systems Architecture* 106 (2020), 101713.

[263] Xiao Zeng, Kai Cao, and Mi Zhang. 2017. MobileDeepPill: A small-footprint mobile deep learning system for recognizing unconstrained pill images. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. 56–67.

[264] Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. 2023. Data-centric artificial intelligence: A survey. *arXiv preprint arXiv:2303.10158* (2023).

[265] Chunhui Zhang, Xiaoming Yuan, Qianyun Zhang, Guangxu Zhu, Lei Cheng, and Ning Zhang. 2022. Toward tailored models on private aiot devices: Federated direct neural architecture search. *IEEE Internet of Things Journal* 9, 18 (2022), 17309–17322.
[266] Jun Zhang and Khaled B Letaief. 2019. Mobile edge intelligence and computing for the internet of vehicles. *Proc. IEEE* 108, 2 (2019), 246–261.
[267] Jie Zhang, Xiaolong Wang, Dawei Li, and Yalin Wang. 2018. Dynamically hierarchy revolution: dirnet for compressing recurrent neural network on mobile devices. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3089–3096.
[268] Linfeng Zhang, Jiebo Song, Anni Gao, et al. 2019. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3713–3722.
[269] Qingyang Zhang, Hong Zhong, Weisong Shi, and Lu Liu. 2021. A trusted and collaborative framework for deep learning in IoT. *Computer Networks* 193 (2021), 108055.
[270] Songli Zhang, Weijia Jia, Zhiqing Tang, Jiong Lou, and Wei Zhao. 2022. Efficient instance reuse approach for service function chain placement in mobile edge computing. *Computer Networks* 211 (2022), 109010.
[271] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6848–6856.
[272] Yongmin Zhang, Xiaolong Lan, Ju Ren, and Lin Cai. 2020. Efficient computing resource sharing for mobile edge-cloud computing networks. *IEEE/ACM Transactions on Networking* 28, 3 (2020), 1227–1240.
[273] Yidan Zhang, Zhiyuan Yan, Xian Sun, Wenhui Diao, Kun Fu, and Lei Wang. 2021. Learning efficient and accurate detectors with dynamic knowledge distillation in remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), 1–19.
[274] Lei Zhao, Youtao Zhang, and Jun Yang. 2020. SCA: a secure CNN accelerator for both training and inference. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 1–6.
[275] Pu Zhao, Wei Niu, Geng Yuan, et al. 2021. Brief industry paper: Towards real-time 3D object detection for autonomous vehicles with pruning search. In *2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 425–428.
[276] Tianming Zhao, Yucheng Xie, Yan Wang, Jerry Cheng, Xiaonan Guo, Bin Hu, and Yingying Chen. 2022. A survey of deep learning on mobile devices: Applications, optimizations, challenges, and research opportunities. *Proc. IEEE* 110, 3 (2022), 334–354.
[277] Yue Zheng, Yuhao Chen, Bin Qian, Xiufang Shi, Yuanchao Shu, and Jiming Chen. 2024. A Review on Edge Large Language Models: Design, Execution, and Applications. *arXiv preprint arXiv:2410.11845* (2024).
[278] Hong Zhong, Li Wang, Jie Cui, Jing Zhang, and Irina Bolodurina. 2023. Secure edge computing-assisted video reporting service in 5G-enabled vehicular networks. *IEEE Transactions on Information Forensics and Security* 18 (2023), 3774–3786.
[279] Qihua Zhou, Song Guo, Zhihao Qu, et al. 2021. Octo: INT8 Training with Loss-aware Compensation and Backward Quantization for Tiny On-device Learning.. In *USENIX Annual Technical Conference*. 177–191.
[280] Qihua Zhou, Zhihao Qu, Song Guo, Boyuan Luo, Jingcai Guo, Zhenda Xu, and Rajendra Akerkar. 2021. On-device learning systems for edge intelligence: A software and hardware synergy perspective. *IEEE Internet of Things Journal* 8, 15 (2021), 11916–11934.
[281] Sha Zhou and Lei Zhang. 2018. Smart home electricity demand forecasting system based on edge computing. In *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 164–167.
[282] Wei Zhou, Yan Jia, Yao Yao, Lipeng Zhu, Le Guan, Yuhang Mao, Peng Liu, and Yuqing Zhang. 2019. Discovering and understanding the security hazards in the interactions between iot devices, mobile apps, and clouds on smart home platforms. (2019), 1133–1150.
[283] Yan Zhou, Shaochang Chen, Yiming Wang, and Wenming Huan. 2020. Review of research on lightweight convolutional neural networks. In *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*. IEEE, 1713–1720.
[284] Zhi Zhou, Xu Chen, En Li, Liekang Zeng, Ke Luo, and Junshan Zhang. 2019. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proc. IEEE* 107, 8 (2019), 1738–1762.
[285] Bingzhao Zhu, Masoud Farivar, and Mahsa Shoaran. 2020. Resot: Resource-efficient oblique trees for neural signal classification. *IEEE Transactions on Biomedical Circuits and Systems* 14, 4 (2020), 692–704.
[286] Sha Zhu, Kaoru Ota, and Mianxiong Dong. 2021. Green AI for IIoT: Energy efficient intelligent edge computing for industrial internet of things. *IEEE Transactions on Green Communications and Networking* 6, 1 (2021), 79–88.
[287] Sha Zhu, Kaoru Ota, and Mianxiong Dong. 2022. Energy-efficient artificial intelligence of things with intelligent edge. *IEEE Internet of Things Journal* 9, 10 (2022), 7525–7532.
[288] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. 2018. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8697–8710.

Received July 2023; revised 2024; accepted 2025