A Survey on The Latest Techniques for Implementing Machine Learning and Artificial Intelligence Algorithms on Resource-constrained Embedded Devices

Bharatha B, Chandana B N, Chirag P I, Deepanjali, Govinda Raju M
Dept. of Electronics and Communication Engineering RV College of Engineering

***Abstract*—This comprehensive examination explores the integration of AI into embedded systems with limited resources, with a focus on the development of AI models and algorithms. It examines neural network compression, hardware acceleration techniques, and modern application paradigms to illustrate the challenges of deploying intelligent technology in constrained environments. By reviewing relevant literature to elucidate the benefits and drawbacks of existing approaches and to guide the discussion toward possible future developments, the study emphasizes the crucial role that embedded AI plays in fostering innovation, efficiency, and compatibility within the Internet of Things. Given the increasing integration of machine learning models into computationally limited devices, this study investigates the platforms, optimizations, and techniques for embedding these models into low-resource microcontroller units (MCUs). The research aims to assist the decentralization of network intelligence while emphasizing embedded AI's ability to increase productivity and accelerate digital transformation by offering concepts, taxonomies, principles, and future directions.**

***Index Terms*—constrained, Machine Learning, Artificial Intelligence, Embedded System**

I. INTRODUCTION

Embedded systems (ES) are revolutionizing daily life with applications spanning consumer electronics, gaming, telecommunications, industrial control, automotive, aeronautics, and military sectors. Their development remains a highly relevant topic in both academia and industry, demanding a specialized approach known as Codesign, where hardware and software are developed concurrently to optimize system performance. The push for miniaturization is accelerating the widespread adoption of ES, shaping a future where these systems will be ubiquitous and profoundly impact our lives.

A. Artificial Intelligence

The rapidly developing topic of embedded artificial intelligence (AI) seamlessly integrates intelligent characteristics into commonplace systems and gadgets. These technologies can analyze data locally, make intelligent judgments, and manage complicated activities on their own without the need for cloud computing or distant servers by incorporating AI directly into edge devices, including smartphones, smart home appliances, industrial machinery, and Internet of Things sensors. Sensors for data collection, processors that can run AI algorithms, and networking choices for device-to-device or cloud-based communication are all part of the technological stack that forms the foundation of embedded AI architecture. This field has benefited greatly from advancements in AI models, especially in machine learning and deep learning, which have made algorithms lighter and more effective, allowing for their use on resource-constrained systems. Embedded AI has a number of benefits over typical AI systems, including improved dependability, higher privacy, and decreased latency. Local data processing and decision-making guarantees real-time answers, which makes it ideal for applications like industrial automation, medical devices, and driverless cars that demand quick decisions. Beyond its influence on technology, embedded AI is essential for improving consumer experiences, increasing operational effectiveness, and forming business models. By using embedded AI, businesses establish themselves as leaders in their field and open up new opportunities for competitiveness, innovation, and long-term success. As embedded AI develops further, it will emerge as a major force behind economic expansion, technical breakthroughs, and social improvement. In the future, embedded AI will be smoothly incorporated into daily life, enhancing the intelligence of systems. Apart from its technological influence, embedded AI is essential for developing company strategies, increasing operational effectiveness, and boosting Implementing embedded AI successfully calls for a thorough grasp of the technology as well as a calculated approach to creativity and problem-solving.

B. Machine Learning

Machine learning (ML) has traditionally been associated with high-performance computing environments that leverage powerful GPUs and cloud infrastructure. However, recent advancements in hardware and software optimization techniques have enabled ML to be deployed on resource-constrained embedded systems, which are characterized by limited processing power, memory, and energy. This shift allows AI-driven intelligence at the edge, reducing dependence on cloud computing and enabling real-time decision-making. The adoption of ML in embedded systems brings several advantages, including low latency and real-time processing, which eliminate delays caused by cloud communication—critical for applications such as autonomous vehicles and medical monitoring. Additionally, running ML locally reduces power consumption, making it ideal for battery-operated IoT devices, enhances privacy and security by keeping sensitive data on the device, and improves bandwidth efficiency by minimizing large data transmissions, which is especially useful in remote applications.

C. ML in Resource-Scarce System

Machine learning (ML) in resource-scarce embedded systems focuses on addressing the challenges of implementing intelligent models in devices with limited computational power and memory, such as microcontroller units (MCUs). These systems often struggle with storing essential ML data representations, including thresholds, hyperplanes, and data points, necessitating careful optimization to balance accuracy, efficiency, and resource constraints. Developers tackle these limitations by designing lightweight algorithms, optimizing execution time, reducing power consumption, and improving scalability. A significant trend in this field is the decentralization of intelligence, which enables ML models to run directly on MCUs and field-programmable gate arrays (FPGAs) at the edge. This shift reduces reliance on cloud computing, enhancing real-time processing, system responsiveness, robust- ness, and overall network efficiency, making embedded AI more autonomous and effective for IoT and smart applications. The unique constraints of resource-scarce embedded systems demand innovative approaches to ML implementation. Since ML models typically require significant storage for data representations, developers must optimize algorithms to minimize memory usage while maintaining efficiency and accuracy. This optimization process involves exploring novel machine learning techniques specifically designed for embedded environments, requiring a delicate balance between execution time, memory consumption, power efficiency, accuracy, and scalability. By leveraging optimization techniques and making informed trade-offs, ML models can be effectively deployed in constrained environments without significantly compromising performance or functionality. Furthermore, the push toward embedding ML in resource-limited systems highlights the growing importance of decentralizing intelligence within IoT networks. By integrating ML capabilities directly into end- devices, such as MCUs and FPGAs, system responsiveness and efficiency can be significantly enhanced. This approach reduces dependency on cloud-based or fog-computing infrastructures, enabling more autonomous, robust, and intelligent network operations. As a result, machine learning on embedded systems is transforming IoT applications, allowing for real-time decision-making, improved energy efficiency, and greater system resilience in a wide range of fields, from industrial automation to healthcare and smart infrastructure.

D. ML in FPGA

Machine learning in Field-Programmable Gate Arrays (FP- GAs) offers a unique opportunity to utilize their reconfigurable nature for implementing intelligent models in embedded systems. FPGAs provide flexibility and parallel processing capabilities, which help accelerate the execution of complex machine learning algorithms. By mapping ML tasks onto FPGA hardware, developers can achieve high performance and low latency, making FPGAs a suitable platform for resource- constrained embedded systems. A major advantage of using

FPGAs for machine learning is their ability to customize hard- ware architectures to optimize specific algorithms and neural network models. This customization allows developers to tailor FPGA implementations to the unique requirements of a given ML task, improving efficiency and performance. Additionally, FPGAs can be dynamically reconfigured to adapt to changing computational demands, making them well-suited for real- time applications where responsiveness is critical. Integrating machine learning algorithms with FPGAs also enables the deployment of intelligent edge devices that can process data locally, reducing dependence on cloud-based resources. This localization of ML capabilities enhances data privacy and security by minimizing the transmission of sensitive information over networks. By embedding ML models directly into FPGAs at the edge, developers can create autonomous, efficient, and responsive systems capable of handling real-time data processing requirements.

E. Tiny ML

Tiny Machine Learning (TinyML) is a breakthrough in artificial intelligence (AI) that enables machine learning (ML) models to run on ultra-low-power microcontrollers instead of traditional cloud-based data centers. This advancement significantly reduces energy consumption, enhances data security, and allows real-time AI processing. With applications spanning across smart cities, precision agriculture, industrial automation, healthcare, and wearable AI, TinyML is trans- forming the way AI is deployed in constrained environments. Understanding TinyML Unlike traditional ML, which relies on large computing infrastructure with ample processing power, memory, and cloud storage, TinyML integrates three key components to function efficiently on constrained devices Hardware, Software, Optimized ML Algorithms.

II. CHALLENGES IN DEPLOYING ML IN ES

Deploying machine learning (ML) on embedded systems comes with several significant challenges due to resource constraints, energy limitations, and real-time processing demands. Limited computational power and memory are major hurdles, as embedded devices, such as microcontrollers and battery-powered robots, often have low-power processors with restricted RAM and storage. Running complex ML models, which typically require intensive floating-point operations and large memory allocations, becomes difficult under these constraints. Additionally, high computational and memory demands further complicate implementation, as traditional ML algorithms require significant processing capabilities that low- power embedded systems cannot support efficiently. This limitation necessitates the use of optimized, lightweight models that balance performance and resource efficiency.

Another major challenge is power consumption, especially in battery-operated or energy-harvested systems, where ML models must run efficiently without depleting power reserves too quickly. Deep learning inference and other high-power ML operations significantly drain battery life, making it necessary to develop power-optimized architectures that maintain efficiency while conserving energy. At the same time, real-time execution constraints pose another difficulty, as many embedded applications, such as autonomous vehicles, industrial automation, and medical monitoring, require immediate responses. However, ML models often introduce latency during inference, which can range from 50 to 500 ms. Such delays are unacceptable for time-sensitive applications, requiring developers to implement latency-optimized models that can provide real-time decision- making.

Model size and storage limitations also play a crucial role in ML deployment on embedded devices. Large models require significant storage space and processing power, which many embedded systems lack. Efficient compression techniques, such as quantization and pruning, are necessary to fit models within memory constraints without sacrificing accuracy. Furthermore, data availability and transmission constraints make cloud-based model updates challenging, as embedded devices often operate in remote locations with intermittent connectivity and limited bandwidth. Since ML models rely on large datasets for training and inference, the restricted storage capacity of embedded systems creates difficulties in handling real-time data locally.

Security and robustness are also critical concerns, particularly for mission-critical applications in healthcare, automotive, and aerospace industries. Embedded ML models are vulnerable to adversarial attacks, data tampering, and privacy breaches due to their limited update mechanisms and mass deployment. Ensuring secure execution, encrypted data transmission, and fault tolerance is essential to maintaining system integrity. Additionally, interoperability remains a challenge, as embedded ML solutions must be compatible across various IoT devices with differing computational capabilities, requiring adaptive frameworks and cross-platform support. Lastly, the complexity of ML model deployment and optimization adds another layer of difficulty, as traditional embedded system developers may lack expertise in training and integrating ML models efficiently. Addressing these challenges requires a combination of hardware and software optimizations to balance performance, power efficiency, security, and real-time responsiveness in embedded machine learning applications. TinyML faces several key challenges due to the limited resources of microcontrollers (MCUs). Since these devices have very little memory and processing power, finding the right balance between model size, accuracy, and efficiency is difficult. Power consumption is another major concern because TinyML devices often rely on batteries and connect to multiple sensors. A lack of standardization in the ML ecosystem creates compatibility issues, making it hard to develop models that work across different devices. One possible solution is creating a unified ML compiler for IoT devices. Security and privacy are also important since TinyML systems interact with both cloud and edge computing. Lightweight security measures are needed to protect data without using too much processing power. Additionally, reliability can be a challenge. Factors like aging hardware, tiny manufacturing differences in chips, and even environmental conditions like radiation can cause minor errors that affect performance. Addressing these challenges is crucial to making TinyML systems more efficient, secure, and dependable on resource-constrained devices.

III. TECHNIQUES TO OVERCOME THESE CHALLENGES

One of the techniques to overcome the above challenges is Limited Computational Power and Memory Constraints. Embedded systems, such as microcontrollers and battery- powered devices, have restricted RAM, storage, and processing capabilities, making it difficult to run complex ML models. To address this, model size optimization techniques such as model pruning, quantization, and knowledge distillation are employed. Additionally, using lightweight architectures like MobileNets, SqueezeNet, and EfficientNetLite helps minimize the number of parameters and computation load, ensuring that the models fit within the memory and computational constraints of embedded devices. Next techniques is High Power Consumption. Where ML models, particularly deep learning-based ones, demand high computational power, which directly impacts battery life in energy-constrained embedded devices. To mitigate this, hardware-aware optimizations are implemented by tailoring models for specific hardware. Dynamic model offloading techniques decide whether to process ML tasks on the edge device or offload them to the cloud based on power and network conditions. These techniques collectively help in extending battery life while maintaining inference performance. Real-Time Execution Constraints techniques in embedded applications, including industrial automation, autonomous vehicles, and medical monitoring, require real-time decision-making, making inference latency a critical factor. To ensure low-latency execution, hierarchical distributed learning is used, where computations are divided between edge devices, edge servers, and cloud servers, optimizing processing speed. Model Size and Storage Limitations in Embedded systems typically have limited flash memory and RAM, making it difficult to store large ML models. To address this, techniques such as static compilation and optimized memory storage are used. Instead of dynamically loading ML models at runtime, models are converted into static C source code. The use of simplified ML models like Na¨ıve Bayes, Decision Trees, and Multi-Layer Perceptron (MLP) further helps in minimizing storage requirements while maintaining accuracy. These approaches ensure that models fit within the storage constraints of embedded devices without performance degradation.

Data Availability and Transmission Constraints where Embedded devices deployed in remote locations often have limited or intermittent network connectivity, making it challenging to update ML models or transmit large datasets to the cloud for processing. To overcome this is employed, allowing devices to train models locally and share only model updates instead of raw data, significantly reducing bandwidth usage. These techniques enable efficient model updates while ensuring that devices function independently without relying on constant cloud connectivity. One of the most important techniques is Security and Robustness Concerns for the Embedded ML applications in critical fields like healthcare, automotive, and aerospace require strong security against adversarial attacks, data tampering, and privacy breaches. Secure execution and encrypted data transmission further protect ML models, ensuring reliability in mission-critical applications. Interoperability Challenges in Embedded ML models need to function across a diverse range of IoT devices with varying computational capabilities, making interoperability a significant challenge. To address this, hardware-specific optimizations are implemented, ensuring that ML models are fine-tuned for different processors and microcontrollers. Complexity of ML Deployment and Optimization in Deploying ML models on embedded systems requires expertise in both machine learning and embedded programming, creating challenges for traditional embedded developers. custom C++ pipeline can be used to deploy ML models on hardware like the Thingy 52, ensuring compatibility despite software constraints. These techniques make it easier for developers to integrate ML models into embedded systems without requiring extensive machine learning expertise.

Tiny Machine Learning (TinyML) enables ML models to run on ultra-low-power microcontrollers, bringing AI capabilities to edge devices. To tackle resource constraints, techniques like quantization (reducing numerical precision) and pruning (removing unnecessary model parameters) optimize memory and computation. Neural Architecture Search (NAS) automates model design to enhance efficiency. Low-power microcontrollers with AI accelerators improve performance while consuming minimal energy. Hardware-software co-design ensures optimized execution by aligning both components. Secure bootloaders prevent unauthorized modifications, enhancing system security. Error Correction Codes (ECC) improve reliability by detecting and correcting computational errors. These techniques collectively make TinyML efficient, secure, and scalable for real-world applications.

IV. ARTIFICIAL INTELLIGENCE

The integration of Artificial Intelligence (AI) into embedded systems marks a significant advancement in the field of intelligent technology. Embedded systems, known for their limited resources, play a crucial role in numerous applications, ranging from consumer gadgets to industrial automation. Implementing AI in these systems requires addressing challenges related to restricted computational power, memory, and energy efficiency. This integration employs sophisticated techniques such as neural network compression, hardware acceleration, and algorithm optimization to achieve smart, efficient, and autonomous operations. As AI continues to expand across various industries, successfully embedding AI into these systems is essential for fostering innovation, improving functionality, and enabling the seamless functioning of the Internet of Things (IoT).

A. Current Developments in Embedded AI

The paper provides a comprehensive discussion on the cur- rent state of development in embedded AI from three critical perspectives: hardware acceleration methods, key technologies, and application models. Firstly, the focus on hardware acceleration methods highlights the importance of optimizing hardware components to support the deployment of AI algorithms on embedded devices efficiently. By exploring various hardware acceleration techniques tailored for neural networks, the paper underscores the significance of enhancing computational capabilities and energy efficiency in embedded AI systems. This perspective sheds light on the advancements in hardware design and optimization strategies that enable the seamless integration of AI technologies into resource on-strained devices, thereby improving performance and overall user experience. Secondly, the examination of key technologies in embedded AI emphasizes the essential role of innovative solutions such as model compression, binary networks, and CPU/GPU acceleration algorithms in shaping the future landscape of AI deployment. These key technologies play a crucial role in determining the feasibility and effectiveness of deploying complex and efficient AI models on embedded systems with limited resources. By introducing and analyzing these technologies, the paper highlights the ongoing efforts to develop lightweight models, optimize algorithms, and enhance computational efficiency in embedded AI applications. This focus on key technologies underscores the importance of continuous innovation and optimization to address the challenges posed by resource constraints in embedded AI systems. Lastly, the discussion on application models in embedded AI provides insights into the practical implementation and utilization of AI technologies in various domains. By exploring real-world applications of embedded AI, the paper showcases the diverse range of use cases and the impact of AI deployment on different industries and sectors. This perspective highlights the versatility and adaptability of AI technologies in addressing complex challenges and enhancing decision-making processes in embedded systems. The analysis of application models underscores the growing significance of AI in driving in- novation, efficiency, and performance improvements across different domains, paving the way for the widespread adoption of embedded AI solutions in diverse applications.

B. Deploying AI on Resource-Constrained Devices

The emphasis on efficient algorithms, lightweight models, optimization of hardware acceleration methods, and deployment methods for compatibility in embedded AI systems underscores the critical focus on enhancing the performance and efficiency of AI applications on resource-constrained devices. By prioritizing the development of efficient algorithms and lightweight models, researchers aim to address the increasing demand for portability and energy efficiency in embedded AI systems. This emphasis on lightweight models and efficient algorithms reflects a strategic approach towards optimizing the deployment of AI technologies on various embedded devices, ensuring seamless integration and improved performance in real-world applications. Additionally, the focus on optimizing hardware acceleration methods and deployment strategies highlights the importance of ensuring compatibility and effectiveness in deploying AI solutions on resource-constrained systems. By fine-tuning hardware acceleration methods and deployment approaches, researchers aim to overcome the challenges posed by limited resources and enhance the overall usability and performance of embedded AI systems [6]. Furthermore, the emphasis on efficient algorithms, lightweight models, optimization of hardware acceleration methods, and deployment strategies underscores the ongoing efforts to drive innovation and efficiency in embedded AI systems. By prioritizing the development of lightweight models and efficient algorithms, researchers aim to strike a balance between performance and resource efficiency in embedded AI applications. This strategic focus on optimization at both the algorithmic and hardware levels reflects a comprehensive approach towards maximizing the capabilities and compatibility of AI technologies in resource-constrained environments. The emphasis on efficient algorithms, lightweight models, and optimized hardware acceleration methods highlights the importance of continuous innovation and optimization in ensuring the successful deployment and integration of AI solutions in embedded systems, paving the way for enhanced performance and usability in diverse applications.

C. Optimizing Efficiency and Compatibility in Embedded AI Systems

The emphasis on efficient algorithms, lightweight models, optimization of hardware acceleration methods, and deployment methods for compatibility in embedded AI systems underscores the critical focus on enhancing the performance and efficiency of AI applications on resource-constrained devices. By prioritizing the development of efficient algorithms and lightweight models, researchers aim to address the increasing demand for portability and energy efficiency in embedded AI systems. This emphasis on lightweight models and efficient algorithms reflects a strategic approach towards optimizing the deployment of AI technologies on various embedded devices, ensuring seamless integration and improved performance in real-world applications. Additionally, the focus on optimizing hardware acceleration methods and deployment strategies highlights the importance of ensuring compatibility and effectiveness in deploying AI solutions on resource-constrained systems. By fine-tuning hardware acceleration methods and deployment approaches, researchers aim to overcome the challenges posed by limited resources and enhance the overall usability and performance of embedded AI systems. Furthermore, the emphasis on efficient algorithms, lightweight models, optimization of hardware acceleration methods, and deployment strategies underscores the ongoing efforts to drive innovation and efficiency in embedded AI systems. By prioritizing the development of lightweight models and efficient algorithms, researchers aim to strike a balance between performance and resource efficiency in embedded AI applications. This strategic focus on optimization at both the algorithmic and hardware levels reflects a comprehensive approach towards maximizing the capabilities and compatibility of AI technologies in resource-constrained environments. The emphasis on efficient algorithms, lightweight models, and optimized hardware acceleration methods highlights the importance of continuous innovation and optimization in ensuring the successful deployment and integration of AI solutions in embedded systems, paving the way for enhanced performance and usability in diverse applications.

D. Key Technologies Shaping Future Deployment in Embedded AI

The introduction of key technologies such as model compression, binary networks, and CPU/GPU acceleration algorithms plays a pivotal role in shaping the future deployment of complex and efficient models and algorithms in embedded AI systems. Model compression techniques aim to maintain model accuracy while reducing its size to accommodate the limited resources of embedded devices, ensuring optimal performance without compromising efficiency. Binary networks further enhance model compression by minimizing the model size while improving accuracy, paving the way for deploying sophisticated AI models on resource-constrained platforms. Additionally, the integration of CPU/GPU acceleration algorithms optimizes hardware support for AI algorithms, enhancing computational efficiency and enabling the deployment of advanced AI applications on embedded systems with improved performance and responsiveness. These key technologies not only address the challenges of deploying AI on resource-constrained devices but also set the stage for the development of more intricate and effective models and algorithms in the embedded AI landscape. By focusing on model compression, binary networks, and CPU/GPU acceleration algorithms, re- searchers aim to overcome limitations in hardware resources and algorithm complexity, thereby enabling the deployment of cutting-edge AI solutions on embedded devices. The strategic implementation of these key technologies underscores a concerted effort to drive innovation and efficiency in embedded AI systems, ensuring compatibility, performance, and scalability in deploying complex AI models and algorithms on diverse resource-constrained platforms.

V. CONCLUSION

In conclusion, integrating machine learning into resource constrained embedded systems is a challenging yet promising frontier in intelligent computing. This integration requires careful attention to performance optimization, memory management, power efficiency, and system scalability to realize the potential of decentralized intelligence fully. Addressing these challenges through innovative algorithm designs, platform specific optimizations, and a holistic approach to system health and flexibility can enable seamless, intelligent decision making at the network edge, improving data privacy, security, and efficiency in IoT ecosystems. Moreover, the integration of AI into embedded systems holds significant promise for enhancing efficiency and intelligence across various industries. This re- view highlights the critical importance of developing efficient algorithms and lightweight models to overcome the limitations of resource limited embedded devices, alongside optimizing hardware acceleration methods. Substantive progress has been made in deploying AI on embedded systems by focusing on model compression, algorithm optimization, and hardware support. However, further advancements are essential to refine these technologies and ensure compatibility, paving the way for embedded AI to play a pivotal role in driving digital transformation and innovation across diverse sectors.

REFERENCES

[1] Angela-Tafadzwa Shumba,Teodoro Montana, Ilaria Sergi, Luca Fache, Massimo De Vittorio, Luigi.Patron, "Embedded Machine Learning To- wards a Low-Cost Intelligent IoT edge" IEEE(2022)

[2] Fateh Boutekkouk, Ridha Mahalaine, Zina Mecibah,Saliha Lakhdari, Ramissa Djouani and Djalila Belkebir "Intelligent Embedded Software New Perspectives and challenges" IntechOpen

[3] EleftheriosBatzolis , Eleni Vrochidou and George A. Papakostas "Ma- chine Learning in

Embedded Systems Limitations Solutions and Future Challenges" IEEE (2023)

[4] Fabianjust,Chiaraghinami,Janzbinden And Maxortizcatalan "Deploy- ment of Machine Learning Algorithms on Resource-Constrained Hard- ware Platforms for Prosthetics" Reasearch Article(2024)

[5] Sunwoong Joo,Attila Dusnoki,Martyn Bliss Ben Duckworth,Nicolas Scotto Di Perto Marko Fabo,Gabor Loki,Daniel Vince,Akos Kiss and Cheul-Hee-Hahm "A Memory-aware Performance Optimization of Ten- sor Programs for Embedded Devices" IEEE(2020)

[6] Borrego-Carazo, Juan, David Castells-Rufas, Ernesto Biempica, and Jordi Carrabina. "Resource-constrained machine learning for ADAS: Asystematic review." IEEE (2020)