Journal of Information Systems Engineering and Management
2025, 10(62s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

On-Device Intelligence: Local Processing Architecture for Mobile Computing Systems

Divya Jain

Independent Researcher, USA

ARTICLE INFO

Received: 06 Oct 2025

Revised: 15 Nov 2025

Accepted: 25 Nov 2025

ABSTRACT

Edge intelligence fundamentally transforms mobile computing architectures by relocating artificial intelligence processing from centralized cloud infrastructure to distributed mobile devices. Traditional cloud-dependent systems encounter inherent limitations, including network latency that precludes real-time interaction, privacy vulnerabilities arising from centralized data aggregation, and accessibility constraints in connectivity-limited regions. The architectural transition to on-device processing addresses these challenges through specialised neural processing hardware implementing highly parallel computational structures optimised for matrix operations and convolutions characteristic of deep learning workloads. This paper makes four primary contributions to understanding edge intelligence architectures. First, systematic examination of specialised neural processing hardware reveals that memory hierarchy design, rather than computational throughput, often determines overall energy efficiency for mobile neural inference. Second, analysis of federated learning augmented with differential privacy mechanisms demonstrates that collaborative model development need not compromise data sovereignty while revealing fundamental trade-offs between privacy strength and model utility. Third, detailed examination of healthcare monitoring, accessibility technologies, and personalized recommendation systems characterizes distinct requirements for local processing, providing quantitative benchmarks: healthcare applications require latency below 100ms for safety-critical detection, accessibility applications demand processing under 20ms to avoid perceptual lag, and personalization systems tolerate latency up to 500ms. Fourth, adaptive hybrid architectures dynamically partition computation between device and cloud, incorporating neural architecture search and once-for-all training paradigms that enable efficient deployment across heterogeneous hardware platforms. The convergence of hardware specialisation, algorithmic optimisation, and privacy-preserving architectures establishes edge intelligence as a viable alternative to cloud-centric artificial intelligence systems.

Keywords: Edge Intelligence, On-Device Processing, Federated Learning, Neural Processing Units, Model Compression, Privacy-Preserving Computing

Introduction

The proliferation of AI in mobile applications has traditionally relied on cloud infrastructure, where compute-intensive inference tasks execute on remote servers. This centralized architecture creates fundamental constraints that limit real-time applications. Network latency in cloud-based inference systems introduces substantial delays - typically 300-600 milliseconds for round-trip communication,

Journal of Information Systems Engineering and Management
2025, 10(62s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

with additional processing overhead extending total response times based on network conditions and server load [1]. For applications requiring human-perceptible real-time interactions, such as augmented reality (requiring <20ms latency), voice assistants (requiring <300ms for natural conversation), and autonomous navigation systems (requiring <100ms for safety), these delays degrade user experience and render certain time-critical applications impractical.

Beyond latency considerations, the very transmission of data to remote servers creates fundamental privacy vulnerabilities. Cloud-based AI architectures require raw sensor data, biometric information, location traces, and behavioural patterns uploaded to external infrastructure. The resultant central collection of such information becomes a target for security breaches, wherein each element being transmitted may result in the sharing of sensitive personal information. Inherently, users are required to entrust third-party service providers with intimate details about daily activities, health status, and communication patterns. Continuous connectivity requirements further reduce accessibility, particularly in areas with limited network infrastructure or during the disruption of connectivity, thereby excluding large user populations from leveraging AI [1]. The distributed nature of mobile sensing environments - from vehicular networks to personal devices conflicts fundamentally with centralized processing paradigms requiring continuous server connectivity.

The regulatory landscape increasingly clashes with centralized processing procedures. Laws across the protection of personal data focus on consumer control, data minimisation, and territorial sovereignty. The movement of data to centralized facilities, especially across jurisdictional borders, makes it more difficult to follow requirements that require sensitive data to remain within certain geographic regions or under direct user control. The cloud-centric model thus generates tension between technological capability and legal demand in data governance.

The key gap that edge intelligence aims to bridge relates to a fundamental tension between advanced artificial intelligence capabilities and operational constraints of mobile deployment. Traditional cloud architecture assumed that meaningful neural network inference, especially for deep networks having high parameter count required computational resources, memory bandwidth, and power budget that go beyond mobile device capabilities. Early mobile processors lacked specialised hardware to perform parallel mathematical operations that are characteristic of neural computation. It meant that the roles of inference had to be delegated to remote servers with dedicated accelerating hardware. However, concurrent advances in specialised hardware architecture and algorithmic optimisation techniques have structurally invalidated this assumption [2].

As one might imagine, the problem of learning from decentralized data sources while preserving privacy has been a driver for research into decentralized learning architectures. Traditional machine learning techniques call for data centralisation on dedicated servers, which is both impractical for many reasons and raises several privacy concerns. Modern mobile devices can generate copious volumes of data that could be used to enhance model performance, but transmission of this information to centralized servers raises very important privacy concerns while consuming large network bandwidth. Federated learning architectures overcome these limitations by training models in a collaborative manner across distributed devices without data centralisation, improving the global model while keeping all data local to the device [2].

This paper makes four primary contributions to understanding edge intelligence architectures. First, comprehensive architecture analysis provides a systematic examination of specialised neural processing hardware designs, contrasting parallel spatial architectures with sequential temporal approaches and analyzing their energy efficiency characteristics. The analysis reveals that memory hierarchy design, rather than computational throughput, often determines overall energy efficiency for mobile neural inference. Second, privacy-preserving framework evaluation analyzes federated learning augmented with differential privacy mechanisms, demonstrating that collaborative model development need not compromise data sovereignty. The evaluation reveals the fundamental trade-off between privacy strength through epsilon-delta parameters and model utility, providing guidance for

practitioners balancing these competing objectives. Third, the domain-specific implementation study provides a detailed examination of healthcare monitoring, accessibility technologies, and personalized recommendation systems, characterizing the distinct requirements and advantages of local processing for latency-critical and privacy-sensitive applications. The analysis provides quantitative benchmarks: healthcare applications require latency below 100ms for safety-critical detection, accessibility applications require processing under 20ms to avoid perceptual lag, and personalization systems can tolerate latency up to 500ms. Fourth, hybrid computational strategy framework presents adaptive architectures that dynamically partition computation between device and cloud, analyzing the multi-dimensional trade-off space including latency, power consumption, accuracy, and privacy. The framework incorporates neural architecture search and once-for-all training paradigms that enable efficient deployment across heterogeneous hardware platforms.

Neural Processing Architecture for Mobile Devices

Specialised Hardware Foundations

Mobile neural processing relies on dedicated hardware units designed specifically for parallel mathematical operations characteristic of artificial intelligence workloads. Unlike general-purpose processors optimized for sequential instruction execution, neural processing units implement highly parallel architectures that leverage the mathematical structure of neural networks. This architectural divergence reflects fundamental differences in computational requirements, where conventional processors handle diverse instruction types with complex control logic, while neural inference demands repetitive arithmetic operations across massive data arrays. The problem of energy consumption in deep neural network processing is not simply a result of computational requirements but also involves moving data around in the memory hierarchies; for instance, accessing external memory consumes many orders of magnitude higher power than arithmetic operations per se [3].

These specialised processors perform basic matrix operations and convolutions - the building blocks of deep learning with much higher energy efficiency compared to traditional processing units. Convolutional operations, being dominant in vision-based networks, exhibit a natural parallelism where the same filter operations apply across spatial dimensions in parallel. This dataflow processing paradigm optimises energy consumption by structuring computation to maximise reuse of data fetched from memory, minimising expensive off-chip memory accesses through carefully designed hierarchies of on-chip buffers. Spatial architectures lay out the processing elements as arrays that process many points in parallel, leveraging neural network operations' abundant parallelism while following data locality, reducing memory access energy [3].

Architectural design favours throughput for certain types of operations while accepting reduced flexibility. This could let the designs perform multiply-accumulate operations - the most common computational pattern in neural inference simultaneously. Memories receive a similar optimisation: their on-chip buffers are designed to minimise the energy-intensive movement of data between the processing elements and main memory. The bandwidth limitations and/or the access energy costs of the memory hierarchy often determine overall energy efficiency for neural inference workloads; hence, memory hierarchies are of primary importance. Temporal architectures execute data sequential processing through their processing engines, while spatial architectures do the processing across arrays of processing elements; each of these offers different trade-offs between flexibility, energy efficiency, and silicon area.

Model Compression Techniques

Deploying neural networks on resource-constrained devices requires systematic model compression approaches that reduce computational and memory requirements, with the preservation of inference

Journal of Information Systems Engineering and Management
2025, 10(62s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

accuracy. Different techniques exist in the compression landscape that address different aspects of model complexity, ranging from the reduction of parameters by pruning to the reduction in numerical precision through quantisation. Modern approaches in compression are all based on the same observation: neural networks are usually quite redundant, where there is a majority of the parameters that contribute negligibly to output predictions, allowing for aggressive compression without catastrophic degradation in accuracy [4].

Quantisation reduces numerical precision from floating-point to fixed-point representations, dramatically decreasing memory footprint and computational requirements. The transition from high-precision floating-point arithmetic to lower-precision fixed-point or integer representations reduces both storage requirements for model parameters and energy consumption for arithmetic operations. Post-training quantisation applies precision reduction to already pre-trained models through calibration procedures that derive optimal scaling factors, while quantisation-aware training integrates quantisation operations into the learning process and therefore allows models to adapt to precision constraints while training instead of retrofitting compression afterward [4].

Network pruning aims at the identification and removal of redundant parameters, based either on magnitude or on sensitivity analysis, which yields sparse network structures where large fractions of weights become zero-valued. In magnitude-based pruning, parameters with small absolute values are removed under the assumption that larger weights contribute more to network predictions. Structured pruning removes whole channels or filters and not individual weights, resulting in regular sparsity patterns. Such patterns can efficiently be exploited by standard hardware without requiring specialised sparse matrix operations. Knowledge distillation is another compression strategy that has recently gained significant attention; it involves training compact student networks to approximate the predictions of larger teacher models. The knowledge transfer happens via soft probability distributions instead of hard classification labels. Many of these compression techniques combine synergistically, with practitioners applying several methods successively to achieve much larger compression ratios than any one technique provides in isolation.

Architecture Component	Traditional Processors	Specialised Neural Processing Units
Execution Pattern	Sequential instruction processing with complex control flow logic	Massive parallel processing through arrays of execution units (1000+ MACs)
Primary Operations	Diverse instruction types with general-purpose arithmetic logic	Matrix multiplications and convolutions optimised for deep learning
Energy Efficiency	Higher power consumption due to general-purpose design overhead	Substantially reduced energy per operation through architectural specialisation
Memory Architecture	Hierarchical cache systems for general workloads	Explicitly managed scratchpad memories sized for neural layer operations
Data Movement	Automatic cache management with external memory dependencies	Minimised off-chip access through optimised on-chip buffer hierarchies
Parallelism Approach	4-16 cores with Limited concurrent execution	Thousands of simultaneous multiply-accumulate operations per cycle
Silicon Utilisation	Complex control logic occupies 30-40% of chip area	Streamlined control enables 80-90% area devoted to computation
Computational Pattern	Flexible instruction execution for varied applications	Single instruction multiple data parallelism at a massive scale

Table 1. Comparative Analysis of Neural Processing Architectures for Mobile Devices [3]

Journal of Information Systems Engineering and Management
2025, 10(62s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Privacy-Preserving Machine Learning Architectures

Federated Learning Framework

Federated learning aims to resolve the fundamental tension between collaborative model improvement and data privacy through an architecture that decouples model training from data centralisation. Instead of centralising the training data, this paradigm of distributed learning trains models on local devices while aggregating only model parameters. The central coordination server never sees raw data but gets parameter updates that reflect learned patterns from the local datasets. This architectural vision fundamentally restructures the conventional workflow of machine learning, in which data collection, storage, and model training occur in centralized facilities under single organisational control [5].

The training protocol runs iteratively through a series of coordinated steps in which local computation and global aggregation are done. Devices download a global model, train it on their private data, compute parameter gradients, and upload these gradients to the coordination server. Communication between the coordination server and participating devices follows an alternation pattern: local training phases, in which a device independently optimises model parameters using its local dataset; and global aggregation phases, where the updates at the coordination server coming from the participating devices are combined. The updates are combined by weighted averaging according to local dataset sizes and yield an improved global model that is to be redistributed. This federated averaging considers heterogeneity in local dataset sizes, allowing devices with more training examples to have a proportional influence on the global model updates [5].

The architecture enables learning from distributed data sources while maintaining strict data locality and thus addresses some fundamental privacy concerns of centralized machine learning systems. However, statistical heterogeneity across participating devices creates significant challenges for federated optimisation algorithms. For instance, due to diverse user populations, device usage patterns, or geographical variations, local data distributions can be non-identically distributed. The degree of heterogeneity directly influences the convergence behaviour; increased distributional divergence increases the number of communication rounds necessary to achieve performance comparable to centralized training. In addition, communication constraints make federated learning deployment challenging, especially in mobile environments where the limitations of network bandwidth, intermittent connectivity, and device availability result in asynchronous participation patterns. Advanced aggregation strategies take these practical constraints into account through client selection mechanisms that explicitly trade off statistical diversity against communication costs [5].

Differential Privacy Mechanisms

Federated learning alone cannot guarantee privacy; gradient information may leak details about training data through various attack vectors. Model updates transmitted during federated training encode information about the training examples used to compute those updates, creating vulnerability to reconstruction attacks. Differential privacy strengthens the federated learning framework through the injection of calibrated noise into parameter updates before transmission. The distribution of noise is designed with care to mask the contribution of any individual while preserving aggregate statistical patterns across many devices [6].

The privacy guarantee is mathematically formalised through epsilon-delta parameters that quantify information leakage bounds. An adversary observing all transmitted updates cannot determine whether any specific data point participated in training beyond probabilities bounded by these privacy parameters. The noise magnitude has to balance privacy strength against model utility: larger noise provides stronger privacy but degrades model performance through reduced gradient accuracy. A fundamental challenge in the implementation of differential privacy is the trade-off between privacy

Journal of Information Systems Engineering and Management
2025, 10(62s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

protection and model accuracy, where stronger privacy guarantees require larger noise additions that obscure the true gradient direction.

Adaptive mechanisms adjust this trade-off depending on data sensitivity and model requirements. The composition properties of differential privacy enable reasoning about privacy degradation over multiple training iterations, as each gradient computation consumes part of the total privacy budget. Advanced accounting methods track the overall privacy expenditure over training epochs, thus allowing practitioners to determine how many iterations are still feasible within acceptable privacy bounds. The convergence analysis of differentially private stochastic gradient descent has shown that noise addition changes the optimization landscape and may require an adaptation of learning rates, batch sizes, or gradient clipping thresholds to maintain stability while preserving the privacy guarantees [6].

Framework Aspect	Federated Learning	Differential Privacy Enhancement
Data Locality	Training occurs locally on individual devices	Noise injection masks individual contributions
Information Transmitted	Model parameter updates and gradients only	Calibrated noisy parameter updates
Central Server Role	Aggregates updates through weighted averaging	Receives privacy-protected parameter updates
Privacy Mechanism	Architectural separation of data and training	Mathematical guarantees through epsilon-delta bounds
Vulnerability	Gradient information may leak training details	Composition properties track cumulative privacy loss
Statistical Challenge	Heterogeneous local data distributions	Trade-off between noise magnitude and accuracy
Communication Pattern	Iterative local training and global aggregation	Additional noise computation before transmission
Attack Resistance	Partial protection through update aggregation	Formal bounds on membership inference attacks
Optimisation Impact	Convergence is affected by distributional divergence	Noise alters gradient directions and training dynamics
Privacy Guarantee	Data never leaves local devices	The adversary cannot determine specific data participation

Table 2. Privacy-Preserving Learning Framework Characteristics [5, 6].

Domain-Specific Implementations

Healthcare Applications

Local processing architectures are instrumental in meeting the peculiar requirements that characterise healthcare data handling in medical monitoring applications. In fact, continuous physiological signal analysis, such as cardiac rhythm monitoring, glucose trend prediction, and respiratory pattern assessment, requires real-time inference on sensitive health data while strictly adhering to healthcare regulations on information privacy. On-device processing eliminates the

Journal of Information Systems Engineering and Management
2025, 10(62s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

transmission of protected health information, thereby simplifying regulatory compliance while offering immediate clinical decision support. The latest integration of machine learning into healthcare data analytics has presented unprecedented opportunities for disease prediction and patient monitoring, whereby the analysis of large-scale health records, social media health communities, and personal monitoring devices can facilitate early detection of medical conditions [7].

The combination of machine learning into healthcare monitoring extends beyond individual patient data to encompass population-level analysis of health trends and disease patterns. Healthcare communities generate significant volumes of data through patient discussions, symptom reports, and treatment reports shared across online platforms. Mining these unstructured data resources provides insights into disease progression patterns, treatment effectiveness, and emerging health concerns that might not be immediately obvious through traditional medical channels. The challenge lies in extracting meaningful clinical insights from noisy, heterogeneous data sources while maintaining patient privacy and ensuring the clinical validity of derived predictions [7].

Disease prediction systems observe distinctive machine learning algorithms to analyse patient data in order to predict health outcomes. Some approaches using classification techniques identify patients at risk for certain conditions based on past health records, demographic factors, and lifestyle indicators. Feature engineering requires carefully selecting clinically relevant variables associated with the onset of a disease while avoiding spurious associations that may hurt the accuracy of predictions. Time-series analysis of patient data highlights trajectories of disease progression, which enables timely intervention before acute conditions arise. The assessment of predictive models in health applications strongly requires validation against clinical ground truth since false alarms create anxiety and resource utilisation, while missed detections possibly delay timely treatments [7].

Accessibility Technologies

Applications in vision and hearing support depend on low-latency processing, crucially operating within perceptual thresholds for determining natural interaction. The real-time visual scene understanding, recognition of sign language, and speech-to-text conversion must operate within such temporal windows beyond which delays in processing become perceptually noticeable, therefore degrading user experience and reducing practical utility. Local neural networks support these applications operating where network connectivity is not accessible, crucial to enabling accessibility in diverse environments where reliable internet access cannot be assumed. In summary, the evolution of on-device AI has fundamentally transformed accessibility applications by enabling sophisticated assistive technologies, previously operating only from cloud infrastructure, to operate wholly on mobile devices [8].

Visual assistance applications implement computer vision models that enable real-time environmental awareness for users with visual impairments. Scene understanding systems locate objects, read text, recognise faces, and detect obstacles while converting visual information into auditory or haptic feedback. The processing pipeline should operate within stringent latency constraints since disorientation and reduced situational awareness can be the result of delays between visual events and corresponding feedback. This necessitates substantial optimisation in the deployment of vision models on mobile devices in order to achieve real-time performance with efficient battery life, since both continuous camera processing and neural network inference place considerable energy demands on portable devices [8].

Context-Aware Personalisation

Recommendation systems traditionally profile users through centralized analysis of behavioural data, aggregating interaction histories, preference signals, and engagement patterns in remote servers.

Journal of Information Systems Engineering and Management
2025, 10(62s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Edge-based approaches enable personalisation while preserving privacy through architectural innovations that maintain user models locally. Consumer models learn preferences locally, adapting to individual styles without external data collection. This structure supports advanced personalisation without requiring users to entrust external parties with behavioural data - an increasingly critical consideration given growing concerns about surveillance and misuse of data in personalised services. The transition to on-device intelligence for personalisation applications reflects broader recognition that privacy preservation and recommendation quality need not represent conflicting targets.

The technical implementation of edge-based recommendation differs fundamentally from cloud-centric approaches in terms of model architecture and update mechanisms. Local recommendation models observe user interactions - content selections, engagement durations, explicit feedback signals to construct preference profiles entirely on-device. This is done by learning algorithms that adapt to evolving user interests through incremental updates, which incorporate new interaction data without requiring complete model retraining. Recommendation quality is critically dependent upon balancing model complexity against device constraints, given that complex collaborative filtering or deep learning architectures may exceed available memory or computational capacity. On-device personalisation models must operate within stringent resource budgets while maintaining prediction accuracy comparable to cloud-based systems that have substantially greater computational resources [8].

Application Domain	Processing Requirements	Privacy Considerations	Latency Constraints	Connectivity Dependence
Healthcare Monitoring	Continuous physiological signal analysis for cardiac rhythms, glucose trends, and respiratory patterns	Protected health information must remain on-device for regulatory compliance	Real-time detection of life-threatening conditions requires immediate inference	Independent operation during connectivity disruptions is essential for patient safety
Accessibility Technologies	Visual scene understanding, sign language recognition, speech-to-text conversion	Minimal privacy concerns, but personal data protection is preferred	Processing delays create perceptual lag, degrading user experience	Functionality without network access is crucial for diverse environments
Context-Aware Personalisation	Incremental learning from user interactions, engagement patterns, and preference signals	Behavioural data is highly sensitive, requiring local profile construction	Responsive adaptation to evolving interests through online learning	Local models eliminate dependency on external recommendation servers
Disease Prediction	Analysis of health records, symptom patterns, and population health trends	Patient data mining requires privacy preservation across communities	Predictive models enable proactive intervention before acute episodes	Population analysis may involve distributed data sources

Table 3. Domain-Specific Edge Intelligence Implementation Requirements [7, 8].

Journal of Information Systems Engineering and Management
2025, 10(62s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Technical Challenges and Hybrid Solutions

Resource Constraints

Modern language models containing billions of parameters well exceed the typical mobile memory capacity, which restricts the sophistication of models that can be deployed on a device. There are fundamental constraints imposed by the memory hierarchy in mobile devices on model deployment: limited random access memory severely restricts the model size that can be loaded into memory at any given time, whereas storage bandwidth limits the rate at which parameters can be streamed from storage during inference. Power consumption continues to be one of the most critical constraints, as continuous neural processing can significantly affect battery life, especially for always-on applications, which may need constant inference. The selection between cloud-based and on-device inference is complex and includes trade-offs across many dimensions, including latency, power consumption, accuracy, and cost. Empirical evaluation shows that neither strategy universally dominates; the best way to deploy DNNs strongly depends on application requirements, network conditions, and device capabilities.

Platform fragmentation complicates the deployment across the heterogeneous mobile ecosystem, in which diverse hardware architectures call for model variants optimised for different processor capabilities. This leads to a fragmented landscape created by the proliferation of different neural processing units from various manufacturers, where models optimised on one hardware platform may be suboptimal in performance on others. The relative study of cloud-versus-on-device inference indicates that network latency is often the dominant contributor to end-to-end response time in cloud-based approaches. Although cloud infrastructure offers access to powerful graphics processing units capable of conducting inference very fast, the overhead of data transmission to remote servers often exceeds the computation time saved through the use of faster processing. On-device inference, on the other hand, eliminates network communication overhead but operates within tightly constrained computational budgets imposed by mobile processors [9].

The energy implications of deployment choices go beyond mere processing costs to include network transmission energy: cloud-based inference involves the transmission of input data to remote servers and receiving prediction results, which consumes significant energy compared with local computation. Energy cost for network transmission varies depending on data size and network technology, leading to cases where large inputs, such as high-resolution images, require more energy to be transmitted to cloud servers than the cost of processing that input on-device using optimised models. Battery constraints, hence, favour local processing in applications with large input data or frequent inference requests, especially when network conditions demand multiple transmission attempts or extended times to maintain a connection [9].

Adaptive Computational Strategies

Hybrid architectures overcome resource limitations by dynamically partitioning computation between the device and the cloud based on query complexity and resource availability. Simple queries execute entirely locally for minimal latency and maximum privacy, whereas complex requests leverage cloud resources when enhanced accuracy justifies additional latency and privacy costs. In most cases, this partitioning decision entails real-time assessment of multiple factors, including input characteristics, available computational resources, network conditions, and privacy sensitivity of the data being processed. Neural architecture search automates the design of efficient networks for specified hardware targets, exploring architectural variations in order to optimize the accuracy-efficiency trade-off. However, conventional neural architecture search methods are computationally expensive processes due to the need for training thousands of candidate architectures in order to find the optimal designs. This becomes prohibitively expensive when targeting multiple hardware platforms, as exhaustive architecture exploration must repeat for each deployment scenario.

Journal of Information Systems Engineering and Management
2025, 10(62s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

These limitations are addressed by the once-for-all training paradigm, which seeks a unified framework that trains a single comprehensive network able to specialise to diverse deployment constraints. Instead of training separate models for different hardware platforms and accuracy requirements, the approach constructs an elastic network that supports variable depth, width, and kernel sizes. The progressive training procedure teaches the network to function correctly across all possible sub-network configurations, which allows specialised models to be extracted without extra training. This methodology transforms the deployment process from an expensive architecture search, which requires substantial computational resources, to efficient model specialisation by straightforward sub-network selection [10].

Context-adaptive computation adjusts model complexity based on runtime conditions, including available battery, connectivity quality, and task urgency, choosing between different computational strategies. Runtime adaptation is achieved via the elastic network architecture since it is designed to support multiple diverse sub-network configurations for varied computational costs and accuracy characteristics. Hardware-aware specialisation identifies the best possible sub-networks on a given device through evolutionary search, where candidate configurations are evaluated on the target hardware directly by measuring actual latency and energy consumption rather than from theoretical operation counts. This hardware-in-the-loop approach discovers architectures that exploit the features of devices whilst respecting the deployment constraints.

Strategy Component	Cloud-Based Processing	On-Device Processing	Adaptive Hybrid Approach
Computational Resources	Access to powerful graphics processing units	Limited by mobile processor capabilities	Dynamic selection based on query complexity
Latency Factors	Network transmission dominates response time	Eliminates communication overhead	Evaluates network conditions for partitioning decisions
Energy Consumption	Wireless transmission energy exceeds local computation for large inputs	Processing energy constrained by battery budget	Battery state influences execution location
Model Sophistication	Support for large models with billions of parameters	Memory constraints limit model complexity	Simple queries local, complex queries offloaded
Privacy Protection	Data transmission creates exposure vulnerabilities	Complete data locality preserves confidentiality	Encrypted computation protocols for cloud processing
Architecture Optimisation	General models serve diverse users	Hardware-aware specialisation for target devices	Once-for-all training enables multi-platform deployment
Deployment Complexity	Centralized model management and updates	Platform fragmentation requires multiple variants	Elastic networks support variable configurations
Context Adaptation	Static model serving all requests	Fixed architecture for specific hardware	Runtime complexity adjustment based on conditions

Table 4. Hybrid Computational Strategy Characteristics [9, 10].

Journal of Information Systems Engineering and Management
2025, 10(62s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Conclusion

The architectural evolution from cloud-centric to edge-based artificial intelligence represents a fundamental restructuring of mobile computing paradigms rather than incremental optimisation of existing approaches. Traditional centralized architectures assumed computation resources beyond those of any mobile device, which necessitates delegating inference tasks to remote servers. Advances in both specialised neural processing hardware and systematic model compression techniques have invalidated assumptions underlying cloud dependency, which have enabled sophisticated inference directly on mobile devices. Specialised processors that implement highly parallel architectures execute matrix operations and convolutions with substantially improved energy efficiency than general-purpose computation, while the methodologies of compression, including quantisation and pruning, reduce model complexity without catastrophic degradation in accuracy. The technical feasibility of on-device intelligence speaks to practical limitations of cloud systems, including network latency incompatible with real-time interaction requirements, and privacy vulnerabilities inherent in the central collection of data. Privacy-preserving frameworks reveal that the collaborative development of models through federated learning, strengthened by differential privacy, ensures, maintains data sovereignty while allowing population-level learning. Domain implementations across healthcare monitoring, accessibility technologies, and personalised services reveal benefits extending beyond technical performance to encompass regulatory compliance, increased accessibility, and user autonomy. Remaining challenges include memory constraints that limit model sophistication and platform fragmentation that complicates deployment. Each necessitates hybrid architectures that can intelligently partition computation between local and remote resources. Adaptive strategies include neural architecture search and context-aware execution that optimise deployment on heterogeneous hardware whilst balancing accuracy against efficiency. The trajectory towards edge intelligence parallels historical transitions from centralized mainframe computing to distributed personal computing, carrying similar implications for accessibility and the democratisation of computation. Future work on efficient architecture design, secure protocols for hybrid computation, and standardised deployment frameworks will ultimately determine the degree to which intelligence becomes an intrinsic device capability instead of a dependency on remote services.

References

[1] Xuan Zhou et al., "When Intelligent Transportation Systems Sensing Meets Edge Computing: Vision and Challenges," MDPI, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/20/9680

[2] H. Brendan McMahan et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 2017. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a/mcmahan17a.pdf

[3] Vivienne Sze et al., "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," arXiv, 2017. [Online]. Available: https://arxiv.org/pdf/1703.09039

[4] Giosué Cataldo Marinó et al., "Deep neural networks compression: A comparative survey and choice recommendations," ScienceDirect, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231222014643

[5] HUIMING CHEN et al., "Advancements in Federated Learning: Models, Methods, and Privacy," ACM Computing Surveys, 2024. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/3664650

[6] JINGWEN ZHAO et al., "Differential Privacy Preservation in Deep Learning: Challenges, Opportunities and Solutions," IEEE Access, 2019. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8683991

Journal of Information Systems Engineering and Management
2025, 10(62s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[7] MIN CHEN et al., "Disease Prediction by Machine Learning Over Big Data From Healthcare Communities," IEEE Access, 2017. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7912315

[8] Xubin Wang et al., "Empowering Edge Intelligence: A Comprehensive Survey on On-Device AI Models," arXiv, 2025. [Online]. Available: https://arxiv.org/pdf/2503.06027?

[9] Tian Guo, "Cloud-based or On-device: An Empirical Study of Mobile Deep Inference," arXiv, 2018. [Online]. Available: https://arxiv.org/pdf/1707.04610

[10] Han Cai et al., "ONCE-FOR-ALL: TRAIN ONE NETWORK AND SPECIALIZE IT FOR EFFICIENT DEPLOYMENT," arXiv, 2020. [Online]. Available: https://arxiv.org/pdf/1908.09791