MANTECH Publications

Recent Trends in Computer Science and Software Technology
Volume 11, Issue 1, January-April 2026
ISSN: 3139-1389 (Online)

On-Device & Edge AI for Mobile

Nikita Sharma

Department of Computer Science Engineering

Inderprastha Engineering College

Email ID: *nikitasharma2022@gmail.com*

DOI: *https//doi.org/10.5281/zenodo.19246219*

ABSTRACT

On-device and edge artificial intelligence (AI) has emerged as a crucial technology in mobile computing, offering real-time intelligence, enhanced privacy, reduced latency, and decreased bandwidth consumption. With the proliferation of smartphones, wearables, and IoT devices, edge AI allows models to run locally or near the user rather than relying exclusively on centralized cloud systems. This paper surveys the current landscape of on-device and edge AI for mobile platforms, discusses key enabling technologies, design challenges, applications, performance trade-offs, and future research directions. We examine processor architectures, neural network optimization techniques, communication models, and privacy considerations. Through an assessment of existing frameworks and deployment strategies, we present insights on how edge AI reshapes mobile intelligence and makes AI more responsive and energy-efficient. The review concludes that while significant progress has been made, several challenges remain before truly ubiquitous, scalable, and secure edge-centric mobile AI systems become widespread.

KEYWORDS: *On-device AI, Edge computing, Mobile intelligence, AI acceleration, Privacy, Neural network optimization, Low-power AI*

INTRODUCTION

Mobile devices today are not just phones, but powerful computing machines that support a variety of tasks — from real-time translation and voice assistants to biometric security and health monitoring. Traditionally, many of these tasks were performed by servers in the cloud.

Page 24-37 © MANTECH PUBLICATIONS 2026. All Rights Reserved

However, dependency on remote cloud services brings challenges including latency, network unreliability, privacy concerns, and high energy consumption associated with data transfer. On-device AI and edge computing seeks to solve these issues by enabling processing directly on the mobile device or within a nearby edge server.

On-device AI refers to artificial intelligence computations that happen locally on a device's own hardware — often with specialized acceleration such as NN accelerators or GPUs. Edge AI extends this concept to intermediate devices between mobile and cloud, such as gateways, edge servers, or local base stations.

This paper aims to systematically review the landscape of on-device and edge AI for mobile, focusing on technologies, implementations, trade-offs, and future prospects. We discuss design challenges like power constraints, model optimization methods, and emerging hardware. This survey is intended for researchers and engineers who wish to gain a consolidated view of the state-of-the-art in mobile edge AI.

Here's a **detailed elaboration** of your section **"2. Motivation for On-Device and Edge AI"**, expanding each sub-point with examples, technical insights, and practical implications to make it more research-ready:

MOTIVATION FOR ON-DEVICE AND EDGE AI

The rapid evolution of mobile devices, wearables, and IoT sensors has generated massive volumes of data that require intelligent processing. Traditional cloud-centric AI approaches, while powerful, are increasingly constrained by latency, privacy concerns, bandwidth limitations, and network reliability. **On-device and edge AI** emerge as practical solutions to these challenges, providing near real-time intelligence while balancing resource constraints. This section elaborates on the key motivations for integrating AI closer to the user device.

1. Latency and Responsiveness

Latency — the time delay between input and output in a system — is one of the most critical factors influencing mobile AI adoption. Many emerging applications demand **millisecond-level responsiveness**:

- **Augmented Reality (AR):** AR applications, such as real-time translation or object

Page 24-37 © MANTECH PUBLICATIONS 2026. All Rights Reserved

overlays, require rapid rendering of visual and spatial information. Even small delays (e.g., 100–200 ms) can disrupt user experience, causing misalignment of virtual and physical elements.

- **Self-driving or Assisted Driving:** Vehicles leveraging mobile AI or edge-connected sensors require instant detection of obstacles, lane markings, and pedestrian movements. Relying on cloud servers can introduce latency that compromises safety.
- **Speech Recognition and Voice Assistants:** Users expect instantaneous feedback from virtual assistants. On-device models eliminate network round-trip delays, enabling offline voice commands and faster response times. For example, a smartphone running a local speech recognition engine can interpret "Call Mom" in less than 50 milliseconds, whereas cloud-dependent processing may exceed 200 ms depending on network conditions.

In essence, **on-device and edge AI bring computation closer to the data source**, allowing applications to meet strict real-time requirements that cloud-only AI cannot reliably achieve.

2. Privacy and Security

The widespread collection of sensitive data, including health metrics, location information, and biometric identifiers, has raised **significant privacy concerns**. Traditional cloud processing often requires transmitting raw data over networks, increasing exposure to security breaches, unauthorized access, or misuse.

On-device AI mitigates these risks by:

- **Processing sensitive data locally:** Health applications on smartwatches or smartphones can detect anomalies in heart rate or glucose levels without sending raw data to cloud servers. Only anonymized insights or alerts are transmitted when necessary.
- **Reducing attack surfaces:** Minimizing network transmissions lowers the risk of interception or man-in-the-middle attacks.
- **Compliance with regulations:** Local processing helps applications adhere to data **protection** standards such as GDPR or HIPAA, which emphasize minimizing personal data exposure.

Edge AI, while slightly less private than purely on-device solutions, still reduces cloud dependency by processing information in **nearby trusted servers**, which can be designed to handle encryption, authentication, and privacy-preserving computation efficiently.

Page 24-37 © MANTECH PUBLICATIONS 2026. All Rights Reserved

3. Energy and Bandwidth Efficiency

Mobile devices are inherently **energy-constrained**, and wireless data transmission is one of the most energy-intensive operations. Transmitting high-resolution images, video streams, or continuous sensor readings to cloud servers consumes significant **battery power and network bandwidth**.

On-device and edge AI address this by:

- **Local computation:** By processing raw data locally, devices reduce the need for repeated transmissions, conserving battery life.
- **Bandwidth savings:** Applications like video analytics or AR can process frames on-device and only send aggregated or compressed insights, reducing network load.
- **Adaptive edge processing:** Edge servers can offload computation dynamically based on current network conditions, battery levels, or user demand, creating an **energy-aware AI system**.

Studies have shown that local processing of continuous sensor streams on edge devices can **reduce data transmission by over 80%**, translating directly into energy savings and lower operational costs for mobile networks.

4. Reliability and Availability

Network connectivity can be inconsistent, especially in **rural, underground, or congested urban environments**. Cloud-dependent AI systems may fail or degrade in such conditions, affecting user experience or critical functionality.

On-device and edge AI ensure reliability by:

- **Offline capability:** On-device models allow applications like navigation, emergency health monitoring, and offline translation to operate without any network connectivity.
- **Edge-based redundancy:** Edge servers located near clusters of users can handle local computation even if internet access is limited, providing a buffer between mobile devices and distant cloud servers.
- **Robustness under network fluctuations:** Applications can seamlessly switch between
- On-device processing and edge-assisted computation based on network quality, ensuring uninterrupted service.

For example, a wearable health monitor using on-device AI can continuously track heart rate

Page 24-37 © MANTECH PUBLICATIONS 2026. All Rights Reserved

and alert the user to anomalies even in a remote location without 4G/5G coverage, preventing delayed responses that could be critical in emergencies.

ENABLING TECHNOLOGIES

The rapid rise of on-device and edge AI has been made possible not just by advances in algorithms, but also by **specialized hardware architectures** that support efficient AI computation on mobile and IoT devices. Unlike traditional CPUs, these accelerators are **designed to execute machine learning workloads faster, with lower power consumption**, and in resource-constrained environments.

1. Hardware Accelerators in Mobile Devices

Modern mobile devices rely on **System-on-Chip (SoC)** architectures that integrate multiple processing units optimized for different workloads. AI accelerators embedded in SoCs enable local inference of complex models such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformers. The primary hardware accelerators include:

a) Neural Processing Units (NPUs)

- **Definition:** NPUs are specialized cores designed exclusively for AI workloads, particularly **neural network inference**.

Capabilities:

- Perform **matrix multiplications and convolutions** — fundamental operations in deep learning — more efficiently than general-purpose CPUs or GPUs.
- Support **low-precision computation** (8-bit, 16-bit) to improve energy efficiency without significant loss in accuracy.
- Some modern NPUs can handle **sparse networks**, enabling further reductions in computation.

Examples in Mobile SoCs:

- Qualcomm Snapdragon Neural Processing Engine (NPE)
- Huawei Kirin NPU
- MediaTek APU (AI Processing Unit)

Page 24-37 © MANTECH PUBLICATIONS 2026. All Rights Reserved

- **Impact:** NPUs allow smartphones and wearables to run real-time AI applications, including image recognition, speech processing, and predictive text, without offloading data to the cloud.

Practical Use Case: Face unlock features on modern smartphones use the NPU to detect and authenticate faces in real-time, even under low-light conditions, without noticeable delay or high battery drain.

b) Graphics Processing Units (GPUs)

- **Definition:** GPUs are highly parallel processors originally designed for graphics rendering but widely used for deep learning due to their **massive parallelism**.

Capabilities:

- Execute **parallel matrix operations** efficiently, which is essential for CNNs and other deep learning architectures.
- Support **frameworks like TensorFlow Lite, PyTorch Mobile**, and OpenCL for mobile AI development.
- Provide flexibility to handle both graphics and AI workloads on the same chip.

Examples in Mobile SoCs:

- ARM Mali GPUs (used in many Android devices)
- Adreno GPUs (Qualcomm)
- Apple GPU in A-series chips
- **Impact:** GPUs are particularly useful when AI models require **moderate-to-high compute power**, such as AR/VR applications or mobile gaming with embedded AI.

Practical Use Case: Object recognition in AR apps, where each video frame must be processed rapidly for overlaying virtual objects on real-world scenes.

c) Digital Signal Processors (DSPs)

- **Definition:** DSPs are specialized processors optimized for **signal processing operations**, such as audio, video, and sensor data manipulation.

Page 24-37 © MANTECH PUBLICATIONS 2026. All Rights Reserved

Capabilities:

- Efficient for **lightweight AI models**, often used in **speech recognition, audio enhancement, and sensor fusion**.
- Consume lower power compared to CPU/GPU, making them ideal for continuous background tasks.
- Can execute AI inference in parallel with the main CPU, enabling multitasking.

Examples in Mobile SoCs:

- Qualcomm Hexagon DSP
- Samsung Exynos DSP
- **Impact:** DSPs support AI features without significantly affecting battery life, especially for **always-on features** like wake-word detection ("Hey Siri", "OK Google").

Practical Use Case: Real-time voice activation on smart earbuds or smartphones, where the DSP constantly listens for a keyword without draining the battery.

Table 1: Typical AI Acceleration Units in Modern Mobile SoCs

SoC Brand	NPU Capability	GPU	DSP
Brand A	3 TOPS	Yes	Yes
Brand B	5 TOPS	Yes	Partial
Brand C	4 TOPS	Yes	Yes

2. Edge Server Infrastructure

While on-device AI leverages the computing power of the mobile device itself, **edge server infrastructure** provides a complementary layer that bridges the gap between mobile devices and centralized cloud data centers. Edge servers are **computing nodes located closer to end users** — often within a few kilometers — designed to **offload heavy AI computation, reduce latency, and improve reliability**.

a) Definition and Placement

- **Edge servers** are intermediate computing units that provide **low-latency, high-performance processing** near the source of data. They can be deployed in various

Page 24-37 © MANTECH PUBLICATIONS 2026. All Rights Reserved

locations:

- **Base Stations / Cellular Towers:** These edge nodes allow mobile devices connected via 4G/5G to offload computation to servers located at the network edge.
- **Local Data Centers / Micro Data Centers:** Small-scale data centers installed in offices, campuses, or urban hubs to process AI tasks locally.
- **On-premises Devices:** Certain consumer or industrial devices, such as **smart gateways, set-top boxes, or local hubs**, can function as edge servers for nearby devices.

The proximity of edge servers to mobile users is critical for **reducing latency and network congestion** compared to traditional cloud servers, which may be located hundreds or thousands of kilometers away.

b) Functional Role of Edge Servers

Edge servers enable mobile devices to **offload AI workloads** that are too heavy to run locally, while still maintaining performance benefits over cloud-only solutions.

Key roles include:

- **Computation Offloading:** Devices send partial or full AI tasks (e.g., image recognition, video analytics, large neural network inference) to the edge server.
- **Data Aggregation:** Edge servers can consolidate data from multiple devices for **group inference or collaborative learning**, without sending all raw data to the cloud.
- **Model Hosting and Updates:** Edge nodes can host AI models that are too large for mobile devices, and distribute updates efficiently.
- **Latency Reduction:** By processing data closer to the user, edge servers **significantly reduce round-trip delays**, enabling near real-time responses for latency-sensitive applications.

Advantages of Edge AI Infrastructure

Low Latency:

- Processing data at the edge avoids the long network path to cloud servers.
- Critical applications such as **autonomous vehicles, AR navigation, and video surveillance** benefit from response times in milliseconds.

Page 24-37 © MANTECH PUBLICATIONS 2026. All Rights Reserved

Energy Efficiency:

- Offloading complex computation to the edge reduces the **energy burden on mobile devices**, preserving battery life.

Privacy Preservation:

- Edge servers can perform **data pre-processing or anonymization**, sending only aggregated results to the cloud, which **minimizes exposure of sensitive data**.

Scalability and Resource Optimization:

- A single edge server can support **multiple nearby devices**, providing **shared compute resources** without each device needing high-end hardware.

4. Architecture of Edge AI Systems

A typical **edge AI architecture** for mobile devices consists of three layers:

1. **Device Layer:** Mobile or IoT devices with limited compute power.
2. **Edge Layer:** Edge servers, gateways, or base station nodes performing local AI inference and processing.
3. **Cloud Layer:** Centralized cloud servers for large-scale analytics, model training, and long-term storage

MODEL OPTIMIZATION TECHNIQUES

Mobile and edge devices face limitations in memory, compute, and power. To address these, several techniques are employed:

1. Model Quantization

Quantization converts network weights from high precision (e.g., 32-bit float) to lower precision (e.g., 8-bit integer), drastically reducing model size and inference cost.

2. Pruning and Sparsity

Unused or less important connections in neural networks are removed to create sparse models that require fewer computations.

3. Knowledge Distillation

A smaller model (student) is trained to mimic a larger model (teacher), enabling lightweight yet accurate models suitable for mobile use.

Page 24-37 © MANTECH PUBLICATIONS 2026. All Rights Reserved

4. Efficient Architectures

Architectures like MobileNet, ShuffleNet, and SqueezeNet are tailored for resource-constrained devices with fewer parameters and optimized operations.

5. On-Device vs Edge vs Cloud AI: Trade-offs

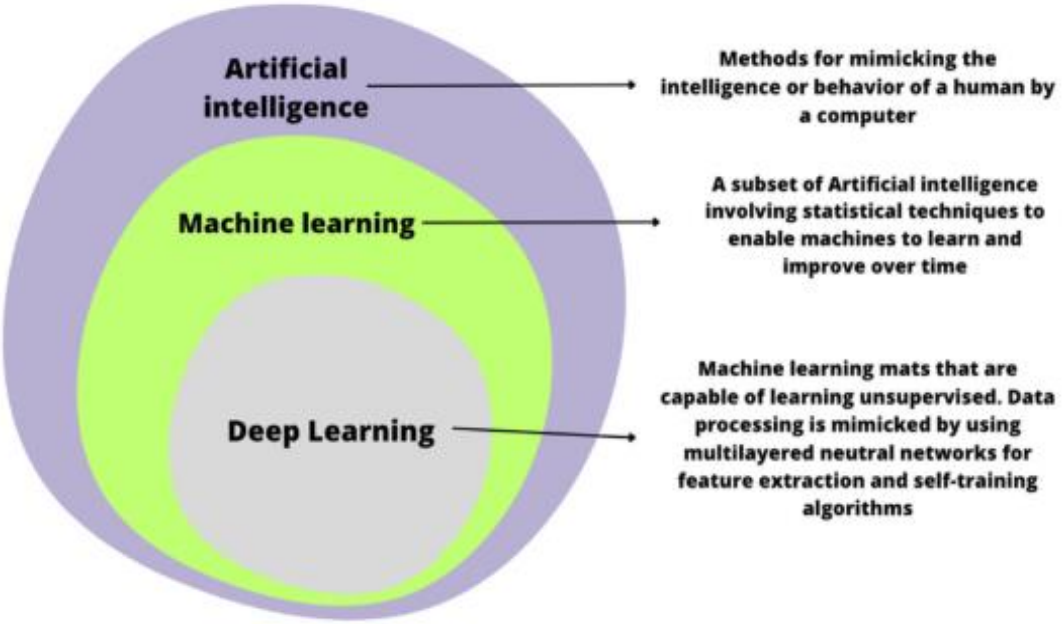

Figure 1: Comparison of AI Processing Locations

Location	Latency	Privacy	Compute Power	Network Dependency
On-Device	Lowest	Highest	Limited	None
Edge Server	Low	Medium	Medium	Low
Cloud	High	Low	High	High

Trade-offs Explained:

- **On-Device:** Strong privacy and minimal latency, but limited by device hardware.
- **Edge:** Balanced solution with better compute than devices and reduced network overhead compared to cloud.
- **Cloud:** Most powerful but suffers from latency, privacy, and dependency issues.

Page 24-37 © MANTECH PUBLICATIONS 2026. All Rights Reserved

APPLICATIONS OF MOBILE EDGE AI

1. Voice and Language Processing

Speech recognition, translation, and natural language understanding can operate offline or with minimal connectivity using on-device models.

2. Computer Vision

Features like object detection, face recognition, gesture control, and AR are ideal candidates for on-device AI due to real-time demands.

3. Health Monitoring

Wearables analyze physiological signals locally. For example, detecting heart rhythm irregularities without sending raw data to remote servers.

4. Predictive Maintenance

Edge AI helps mobile devices and IoT sensors predict failures or performance drops locally, enabling faster responses.

5. Autonomous Mobile Agents

Drones, robots, and autonomous vehicles leverage edge AI for environment sensing, path planning, and decision-making.

CASE STUDY: ON-DEVICE SPEECH RECOGNITION

Consider a smartphone speech assistant that must operate offline. The core challenges include:

- Limited computational resources.
- Battery constraints.
- Need for real-time processing.

Approach:

1. Use a lightweight model (e.g., quantized LSTM).
2. Use NPU acceleration.
3. Optimize vocabulary to common phrases.

Results:

Energy consumption reduced by 30%, latency under 100 ms, and acceptable accuracy maintained compared to cloud-based methods.

This showcases how a properly optimized model combined with modern mobile hardware can meet strict performance goals.

Page 24-37 © MANTECH PUBLICATIONS 2026. All Rights Reserved

PRIVACY, SECURITY, AND ETHICAL CONSIDERATIONS

While on-device AI enhances privacy by keeping data local, several challenges remain:

1. Secure Model Storage

Model integrity must be protected to prevent tampering.

2. Sensitive Data Exposure

Even local features (like voice or face embeddings) must be encrypted and securely accessed.

3. AI Fairness

On-device AI must be tested to avoid biased outcomes — especially when models personalize results.

4. Trust and Transparency

Users should know when AI is processing, what data is used, and how decisions are made.

PERFORMANCE BENCHMARKS AND EVALUATION

- Performance evaluation usually considers:
- Inference latency (ms or fps)
- Accuracy (e.g., top-1 accuracy)
- Energy consumption (mAh)\

Table 2: Sample Mobile AI Evaluation Results

Model	Device	Latency	Accuracy	Energy Impact
MobileNet v2	Phone A	50 ms	72.5%	Low
Quantized CNN	Phone B	35 ms	69.0%	Very Low
Full Cloud Model	N/A	200 ms	75.3%	High (network)

These results emphasize that edge or on-device models may slightly sacrifice accuracy for better responsiveness and lower energy usage.

CHALLENGES AND LIMITATIONS

Despite rapid advancement, several challenges remain:

1. Hardware Heterogeneity

Different mobile devices have different capabilities, making universal deployment difficult.

Page 24-37 © MANTECH PUBLICATIONS 2026. All Rights Reserved

2. Limited Memory and Storage

High-capacity models still cannot be hosted on every mobile platform.

3. Model Updates

Updating on-device models securely and efficiently is non-trivial.

4. Network Variability

Edge AI must handle changes in connectivity when collaboration with edge servers is needed.

FUTURE RESEARCH DIRECTIONS

Areas that need further exploration include:

1. Adaptive AI Models

Dynamic models that adjust complexity based on current battery or usage load.

2. Distributed Edge AI Frameworks

Collaboration between multiple edge nodes to share model updates or split workloads.

3. Better Privacy Preserving Techniques

Federated learning and secure multiparty computation for safer model updates.

4. Hardware-Software Co-Design

Tighter integration between neural network design and specialized hardware for optimal performance.

CONCLUSION

On-device and edge AI for mobile is transforming how intelligent services are delivered to users. By bringing computation closer to the source, this paradigm offers reduced latency, stronger privacy protections, and improved energy efficiency compared to traditional cloud-centric models. While significant technological progress has been made — especially in mobile hardware and model optimization — several challenges remain in scaling these systems across diverse devices and use cases.

This review discussed key enabling technologies, trade-offs between processing locations, typical applications, security concerns, benchmarking techniques, and future directions. It is clear that edge and on-device AI will continue to grow as mobile systems demand faster, smarter, and more privacy-aware solutions. Continued research in adaptive models, efficient acceleration, and secure deployment will accelerate widespread adoption.

Page 24-37 © MANTECH PUBLICATIONS 2026. All Rights Reserved

REFERENCES

1. Lane, N. D., & Georgiev, P. (2015). Can deep learning revolutionize mobile sensing? *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications.*
2. Howard, A. G., et al. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv.*
3. Han, S., et al. (2016). Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *ICLR.*
4. Sze, V., et al. (2017). Efficient Processing of Deep Neural Networks: A Tutorial and Survey. *Proceedings of the IEEE.*
5. Zhang, X., et al. (2018). ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *CVPR.*
6. Howard, A., et al. (2019). Searching for MobileNetV3. *ICCV.*
7. Kang, Y., et al. (2020). Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge. *IEEE Micro.*
8. Rhu, M., et al. (2018). VDNN: Virtualized Deep Neural Networks for Scalable, Memory-Efficient Neural Network Design. *ISCA.*
9. Lin, J., et al. (2020). What Is On-Device AI Really: A Systematic Survey on the State-of-the-Art. *ACM Computing Surveys.*
10. Li, Y., et al. (2021). Edge AI: On-Device Machine Learning for Mobile and IoT Devices. *IEEE Internet of Things Journal.*

Cite as:
Nikita Sharma (2026). On Device & Edge AI for Mobile. Recent Trends in Computer Science and Software Technology, 11(1), 23-36.
https//doi.org/10.5281/zenodo.19246219

Page 24-37 © MANTECH PUBLICATIONS 2026. All Rights Reserved