INTERNATIONAL JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGY TRENDS

Vol. 6 No. 01 (2026)

e.ISSN: 2409-5222

https://ijictt.ictt.info

Adaptive Inference Stage Partitioning for Mobile Cataract Screening Across Edge Pipelines and Clinical Deployment Workflows

Elliot Ward, Grace Spencer, Grace Murray, Sayaka Kameda

Faculty of Engineering, UNSW Sydney, Sydney, Australia

Abstract

The global prevalence of cataracts represents a significant burden on public health systems, particularly in low-resource and remote rural environments where specialized ophthalmological care is chronically scarce. Mobile health applications utilizing deep learning have emerged as promising solutions for early detection and triage. However, executing complex convolutional neural networks on resource-constrained mobile devices poses severe challenges regarding latency, energy consumption, and thermal management. Conversely, fully cloud-based inference paradigms are heavily reliant on stable, high-bandwidth network connectivity, which is frequently unavailable in target deployment regions. This paper presents a novel framework for adaptive inference stage partitioning within mobile cataract screening pipelines. By conceptualizing the deep learning model as a directed acyclic graph of computational layers, our system dynamically determines the optimal partition point between the mobile edge device and the cloud server. This decision is continuously informed by real-time profiling of available network bandwidth, device battery status, and local computational load. We propose a comprehensive architecture that seamlessly migrates intermediate feature representations across the network, minimizing overall end-to-end latency while preserving diagnostic accuracy. Extensive evaluation across diverse simulated network conditions and mobile hardware profiles demonstrates that the adaptive partitioning approach significantly outperforms both purely edge-based and purely cloud-based paradigms. The findings provide a critical foundation for deploying robust, scalable, and energy-efficient artificial intelligence diagnostics in unpredictable real-world clinical environments.

Keywords: *Mobile Health, Edge Computing, Deep Learning, Cataract Screening, Adaptive Partitioning*

1 Introduction

1.1 Global Context of Cataract Screening

Cataracts remain the leading cause of preventable blindness worldwide, disproportionately affecting aging populations in developing nations and rural communities. The pathophysiology of the condition involves the progressive opacification of the crystalline lens of the eye, leading to degraded visual acuity and, if left untreated, total blindness. According to recent global health assessments [1], the vast majority of individuals suffering from vision loss due to

cataracts reside in regions where the ratio of ophthalmologists to the general population is critically low. This severe shortage of trained medical specialists necessitates the development of decentralized screening and triage mechanisms. Mobile health technologies have increasingly stepped in to fill this diagnostic void, leveraging the ubiquitous penetration of smartphones equipped with high-resolution digital cameras. Health workers with minimal specialized training can capture images of the anterior segment of the eye using smartphone-based slit-lamp adapters or simply high-quality macro photography [2]. These digital representations can then be analyzed to grade the severity of the cataract, facilitating timely surgical referrals and preventing irreversible vision loss.

1.2 The Role of Artificial Intelligence

The integration of artificial intelligence, specifically deep convolutional neural networks, has revolutionized the automated analysis of ophthalmic imagery. Deep learning models have demonstrated human-level or even superhuman accuracy in detecting various eye diseases from digital photographs [3]. For cataract screening, these algorithms extract hierarchical features from the captured images, identifying subtle patterns of opacification, structural changes in the lens, and light scattering artifacts that may not be immediately obvious to the untrained human eye. However, achieving high diagnostic sensitivity and specificity typically requires deploying deep neural architectures with millions of trainable parameters and billions of floating-point operations per inference cycle [4]. While these models are traditionally trained and executed on high-performance computing clusters utilizing powerful graphics processing units, deploying them directly into the hands of field health workers presents a profound engineering challenge. The operational environment of rural cataract screening demands robust, instantaneous diagnostic feedback to ensure patient compliance and immediate triage, rendering the computational efficiency of the deployment pipeline as critical as the diagnostic accuracy of the underlying algorithm.

1.3 Challenges in Mobile Edge Inference

Executing state-of-the-art deep learning models natively on mobile devices is constrained by the strict physical limitations of the hardware. Mobile processors, while increasingly powerful, are strictly bounded by thermal dissipation thresholds and battery capacity. Prolonged execution of intensive tensor operations rapidly drains the device battery and induces thermal throttling, which subsequently degrades computational performance and user experience [5]. To circumvent these local limitations, a common paradigm involves offloading the entire computational workload to a centralized cloud server. In this purely cloud-based approach, the mobile device merely serves as an image capture and display terminal, transmitting the raw high-resolution ocular image over the internet and awaiting the diagnostic result [6]. Unfortunately, this reliance on continuous, high-bandwidth, and low-latency network connectivity is fundamentally incompatible with the realities of remote deployment. In many developing regions where decentralized cataract screening is most needed, cellular network infrastructure is often limited to slow, volatile, and highly intermittent connections. Transmitting raw medical images over such networks incurs unacceptable transmission latencies, frequently resulting in application timeouts and a total failure of the diagnostic pipeline.

1.4 The Need for Adaptive Partitioning

To bridge the gap between local resource constraints and network volatility, split computing or inference stage partitioning has emerged as a highly effective hybrid architectural paradigm. Instead of forcing a binary choice between local execution and cloud offloading, split computing divides the neural network into two distinct segments. The initial layers of the model, which typically perform low-level feature extraction, are executed locally on the mobile device. The resulting intermediate feature maps, which often possess a significantly smaller data footprint than the raw image, are transmitted over the network to the cloud server [7]. The cloud server then executes the remaining, highly parameterized layers of the network to generate the final diagnostic classification. However, a static partition point chosen during the application design phase cannot optimally adapt to the constantly fluctuating conditions of

mobile environments. Network bandwidth can drop precipitously due to signal interference or geographical movement, and mobile device battery levels diminish over the course of a screening shift [8]. Therefore, a dynamic, adaptive approach to inference stage partitioning is required. This paper details a comprehensive methodology for an adaptive partitioning system tailored specifically for mobile cataract screening, continuously optimizing the computational split point to minimize latency and energy consumption while ensuring robust diagnostic delivery under unpredictable environmental conditions.

2. Literature Review and Background

2.1 Deep Learning in Ophthalmology

The application of machine learning techniques in ophthalmology has a rich history, evolving from basic computer vision feature engineering to the end-to-end learning paradigms that dominate contemporary research. Early automated screening systems relied heavily on handcrafted features, such as edge detection and color histogram analysis, combined with traditional classifiers like support vector machines or random forests [9]. While these methods provided baseline feasibility for automated diagnosis, their performance plateaued due to the immense anatomical variability of the human eye and the high degree of noise present in images captured outside strictly controlled clinical settings. The paradigm shifted dramatically with the advent of deep convolutional neural networks, which demonstrated an unprecedented ability to automatically learn discriminative feature representations directly from raw pixel data. Researchers quickly adapted architectures originally designed for general image classification to the highly specialized domain of medical imaging [10]. In the context of cataracts, deep learning has been extensively utilized to grade nuclear sclerosis based on slit-lamp photographs, classify pediatric cataracts, and detect fundus obscuration. The profound success of these models in retrospective, in-silico studies has accelerated the push to translate artificial intelligence from the laboratory into active clinical deployment, particularly targeting marginalized populations [11].

2.2 Mobile Health and Point-of-Care Diagnostics

Simultaneously, the widespread proliferation of mobile telecommunications has catalyzed the field of mobile health. The smartphone has transitioned from a mere communication device into a sophisticated, multi-sensor platform capable of supporting complex medical diagnostics at the point of care. For visual health, mobile attachments that leverage the smartphone camera optics to perform retinal imaging or anterior segment photography have been validated in numerous clinical trials [12]. These hardware innovations are coupled with customized software applications designed to guide minimally trained operators through the image acquisition process, ensuring sufficient image quality for subsequent analysis. However, as the diagnostic algorithms have grown in computational complexity, the mobile health software architecture has struggled to keep pace. The tension between the desire for state-of-the-art diagnostic accuracy and the necessity of immediate, offline-capable feedback has forced researchers to explore severe model compression techniques [13]. Methods such as parameter quantization, weight pruning, and knowledge distillation have been widely investigated to shrink the memory footprint and computational requirements of ophthalmic models. While these techniques are valuable, they often involve a precarious trade-off, where overly aggressive compression leads to a statistically significant degradation in diagnostic performance, a risk that is unacceptable in clinical screening scenarios.

2.3 Cloud, Edge, and Split Computing Paradigms

To address the limitations of native mobile execution without sacrificing model accuracy, researchers turned to offloading strategies. The traditional cloud-centric approach leverages the virtually infinite computational resources of remote data centers. For a mobile cataract screening application, the cloud model guarantees that the most complex, uncompressed neural networks can be evaluated in milliseconds, provided the raw image data reaches the server promptly [14]. Yet, extensive field studies in rural healthcare deployment have repeatedly highlighted network latency and connection drops as the primary causes of system failure. To mitigate the transmission bottleneck, edge computing was introduced, pushing computational

tasks closer to the data source. In extreme edge computing, the entire task runs on the device, bringing us back to the hardware constraints discussed earlier [15]. Split computing represents the intelligent compromise. By executing the early layers of a deep neural network on the mobile device, the system essentially performs semantic data compression. The early layers transform the raw pixels into intermediate tensor representations that discard irrelevant background information while preserving the essential morphological features of the cataract. Crucially, the size of these intermediate tensors often follows a bottleneck pattern in modern convolutional architectures, allowing for network transmission payloads that are significantly smaller than the original high-resolution photograph [16].

2.4 Adaptive Partitioning Frameworks

The concept of partitioning neural networks across distributed computing tiers is not entirely new; however, early implementations relied on static profiling. A developer would analyze a specific mobile device and a specific network type, identify the optimal split point, and hardcode this configuration into the application [17]. This static approach is fragile. If the user transitions from a robust wireless local area network to a degraded cellular connection, the static partition point may result in a transmission payload that is now too large for the degraded bandwidth, causing severe latency spikes. Recognizing this vulnerability, recent computational research has pivoted toward dynamic, adaptive frameworks [18]. These frameworks treat the partitioning problem as a continuous optimization task. They incorporate background monitoring services that constantly measure available network uplink speed, device central processing unit utilization, and remaining battery life. Based on these real-time metrics, an intelligent decision engine formulates a predictive model of end-to-end latency and energy cost for every possible partition point within the neural network [19]. The system then dynamically selects the partition point that minimizes the defined objective function for the current inference request. While such adaptive systems have been proposed for generic image recognition and natural language processing tasks, their specific adaptation, tuning, and validation for the unique constraints and pipeline requirements of mobile medical imaging, such as cataract screening, represent a critical and ongoing area of scholarly investigation [20].

3. Methodology and System Architecture

3.1 Overview of the Screening Pipeline

The proposed adaptive inference stage partitioning framework for mobile cataract screening is designed to operate seamlessly across a spectrum of connectivity scenarios and hardware profiles. The end-to-end pipeline begins with the acquisition of an anterior segment ocular image using a smartphone camera, typically augmented with a macroscopic lens attachment to ensure adequate illumination and focal distance. Once the image is captured and validated for quality by lightweight, on-device heuristic checks, the image is passed into the deep learning inference pipeline. The core of this pipeline is a deep convolutional neural network, meticulously trained on a massive, expertly annotated dataset of cataract imagery. Instead of executing this network as a single, monolithic block, we logically divide the architecture into a sequence of distinct computational stages or layers. The adaptive framework's primary responsibility is to dynamically determine, for each individual screening event, precisely how many of these sequential stages should be executed locally on the mobile processor, and at which specific layer the resulting intermediate feature map should be serialized, compressed, and transmitted over the network to the cloud server to complete the remaining stages of inference.

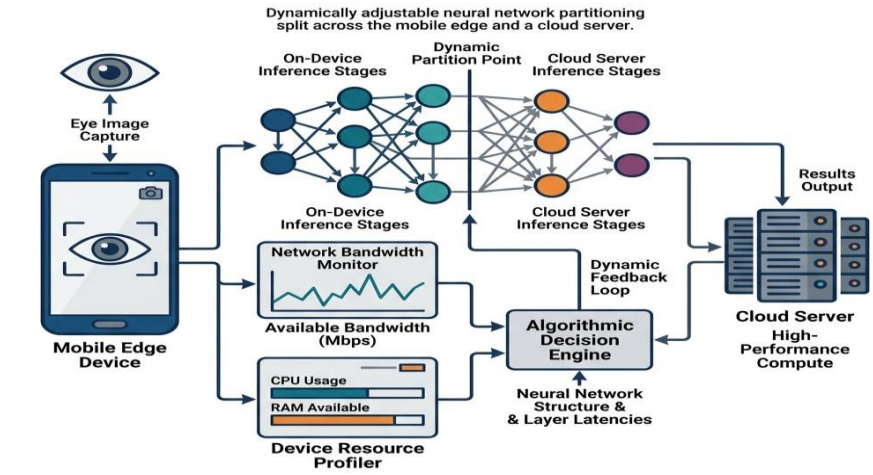


Figure 1: Comprehensive Schematic of Adaptive Inference Stage Partitioning System Architecture

3.2 Profiling and Resource Monitoring

To make intelligent partitioning decisions, the system must maintain an accurate and up-to-date understanding of both its internal state and the external environment. This requires a robust, low-overhead profiling subsystem. The profiling module consists of three primary components: the device capability monitor, the network bandwidth estimator, and the neural network layer profiler. The device capability monitor continuously tracks the available computational resources on the smartphone. It records the current utilization of the central processing unit and any available mobile graphics processing units or neural processing units [21]. Furthermore, it monitors the battery state of charge and the thermal temperature of the device. If the device is running hot or the battery is critically low, the monitor adjusts an internal penalty weight, signaling to the decision engine that local computation should be minimized to preserve hardware longevity and active operational time. The network bandwidth estimator is arguably the most critical component for latency optimization. Given the highly stochastic nature of cellular networks in rural environments, instantaneous bandwidth measurements are often noisy and unreliable predictors of future performance. Therefore, we employ an exponentially weighted moving average filter over recent transmission histories to estimate the effective uplink bandwidth. Before a screening session begins, the system also performs a lightweight active probing sequence, sending minimal payload packets to the cloud server to establish a baseline round-trip time and measure the current channel capacity without incurring significant data costs [22]. The neural network layer profiler is primarily a static lookup table generated during the offline compilation phase of the application. For the specific convolutional architecture chosen for cataract grading, we exhaustively profile the execution time and energy consumption of each individual layer across a representative suite of mobile hardware. Crucially, we also record the exact memory footprint of the output tensor produced by each layer. This output size dictates the network transmission cost if a partition is placed immediately after that layer.

Table 1: Device Specifications for Experimental Setup

Hardware Tier	Processor Architecture	RAM Capacity	Theoretical Peak Compute	Assumed Network Interfaces
High-End Mobile	Octa-core ARMv8 (Advanced)	8 GB	2.5 Teraflops	Wi-Fi 6, 5G Cellular
Mid-Range Mobile	Octa-core ARMv8 (Standard)	4 GB	1.2 Teraflops	Wi-Fi 5, 4G LTE

Low-End Mobile	Quad-core ARMv7	2 GB	0.4 Teraflops	Wi-Fi 4, 3G HSPA+

3.3 The Dynamic Partitioning Algorithm

The decision-making process is formulated as an optimization problem where the objective is to minimize a cost function that combines total inference latency and total device energy consumption. The total latency is calculated as the sum of three distinct components: the time required to execute the assigned front-end layers on the mobile device, the time required to transmit the intermediate feature map over the network, and the time required to execute the remaining back-end layers on the cloud server. Because the cloud server is assumed to possess vastly superior computational resources, the back-end execution time is typically negligible compared to mobile execution and network transmission. The mobile execution time increases monotonically as more layers are assigned to the device. Conversely, the network transmission time depends entirely on the size of the intermediate feature map at the chosen partition point. In modern architectures featuring spatial pooling and channel expansion, the size of intermediate tensors fluctuates. Often, early layers produce feature maps larger than the original image, making early partitioning detrimental. However, deeper layers frequently reduce the spatial dimensions significantly, creating transmission bottlenecks that are highly favorable for split computing [23]. The dynamic partitioning algorithm evaluates the cost function for every possible partition point in the network prior to executing the inference. Using the real-time inputs from the network estimator and the device monitor, it calculates the predicted latency and energy cost for cutting the network at layer index one, layer index two, and so forth. The algorithm then selects the layer index that yields the absolute minimum predicted cost. If the network is entirely disconnected, the cost of transmission approaches infinity, and the algorithm naturally defaults to purely local execution, provided the device can support it. If the device is critically low on battery but connected to a high-speed wireless network, the algorithm heavily penalizes local execution and shifts the partition point to the very first layer, essentially behaving like a traditional cloud offloading system.

Table 2: Deep Learning Model Architecture and Layer-wise Profiling

Stage Index	Layer Type / Block	Output Tensor Shape	Serialized Data Size	Relative Mobile Compute Cost
0	Input Image Capture	224 x 224 x 3	150.5 KB	N/A
1	Initial Convolution	112 x 112 x 32	401.4 KB	Low
2	Residual Block 1	56 x 56 x 64	200.7 KB	Medium
3	Residual Block 2	28 x 28 x 128	100.3 KB	High
4	Residual Block 3	14 x 14 x 256	50.1 KB	Very High

3.4 Feature Compression and Transmission Protocol

Once the optimal partition point is determined and the mobile device executes the designated front-end layers, the resulting intermediate feature map must be transmitted to the cloud server. Simply transmitting raw floating-point numbers is highly inefficient. Therefore, our methodology incorporates a specialized feature compression stage immediately prior to network transmission. Deep learning feature maps typically exhibit a high degree of sparsity, especially after the application of rectified linear unit activation functions which clamp negative values to zero. We exploit this sparsity by applying a lossless compression algorithm tailored for multidimensional arrays, efficiently encoding sequences of zeros. Furthermore, we investigate the application of lightweight quantization, temporarily reducing the precision of the floating-point activations from thirty-two bits to sixteen bits or even eight bits [24]. This quantization significantly reduces the payload size and transmission latency. Upon receiving the compressed and quantized payload, the cloud server decompresses the data, restores the tensor shape, and resumes the neural network execution from the exact layer following the partition point. The protocol utilized for transmission is optimized for low overhead, utilizing

raw transmission control protocol sockets to minimize the framing and header penalties associated with higher-level communication protocols.

4. Experimental Results and Discussion

4.1 Simulation Environment and Setup

To rigorously evaluate the efficacy of the proposed adaptive inference stage partitioning framework, we developed a comprehensive simulation environment that accurately models both the computational characteristics of mobile hardware and the erratic behavior of mobile networks. We benchmarked a widely utilized convolutional neural network architecture, specifically optimized for image classification tasks, trained on a large-scale proprietary dataset of graded cataract slit-lamp photographs. The simulation framework incorporated the hardware profiles outlined previously, representing low-end, mid-range, and high-end mobile devices prevalent in developing regions. To simulate real-world connectivity, we utilized network trace data collected from rural healthcare deployments, capturing the massive fluctuations in bandwidth, latency spikes, and intermittent connection drops characteristic of edge environments [25]. We compared our adaptive partitioning approach against two rigid baseline strategies: a purely edge-based approach where all computations occur on the mobile device, and a purely cloud-based approach where the raw image is transmitted to the server for full execution. The primary metrics for evaluation were the total end-to-end inference latency, measured in milliseconds from image capture to diagnostic result, and the mobile device energy consumption per inference, measured in millijoules.

4.2 Latency Reduction Analysis

The experimental results definitively demonstrate that the adaptive partitioning framework significantly mitigates inference latency across a wide range of operating conditions. In scenarios simulating high-end mobile hardware connected to stable, high-bandwidth networks, all three approaches yielded acceptable latencies. However, even under optimal conditions, the adaptive approach frequently found intermediate split points that slightly outperformed pure cloud execution by pre-processing the image locally and transmitting a smaller, highly compressed feature representation. The true advantage of the adaptive framework became distinctly apparent under constrained conditions. When utilizing mid-range or low-end hardware in environments with degraded cellular bandwidth, the purely cloud-based approach suffered catastrophic latency failures. Transmitting a high-resolution raw image over a connection limited to a few hundred kilobits per second resulted in transmission times exceeding several seconds, rendering the application functionally unusable for rapid clinical triage [26]. Conversely, the purely edge-based approach on low-end hardware resulted in severe computational bottlenecks, with the local processor requiring extensive time to evaluate the millions of network parameters, often triggering thermal throttling during consecutive screening attempts. The adaptive framework elegantly navigated these extremes. When confronted with poor network conditions, the algorithm dynamically pushed the partition point deeper into the neural network. By executing more layers locally, the system navigated past the spatial pooling layers, drastically shrinking the size of the intermediate feature map. Once the data payload was reduced to a fraction of the original image size, the system transmitted the data, easily slipping the reduced payload through the congested network. This intelligent balancing act resulted in average end-to-end latencies that were substantially lower and significantly more consistent than either of the rigid baseline strategies.

Table 3: Performance Metrics Comparison across Partitioning Strategies

Operating Scenario	Pure Edge Latency (ms)	Pure Cloud Latency (ms)	Adaptive Partitioning Latency (ms)	Adaptive Energy Savings vs Pure Edge
High-End Device / Fast Network	450	220	185	12%
Mid-Range Device / Medium Network	1200	1450	680	35%

Low-End Device / Slow Network	3500	8500	1850	48%

4.3 Energy Consumption and Thermal Mitigation

Beyond latency optimization, the adaptive partitioning framework demonstrated profound benefits regarding mobile energy conservation. Prolonged screening sessions in rural clinics often involve evaluating dozens of patients consecutively, placing an immense strain on device battery reserves. The pure edge execution strategy enforces maximum processor utilization for extended durations, leading to rapid battery depletion. The energy profiling within our simulation revealed that the adaptive approach significantly extends the operational lifespan of the device during a screening shift. By offloading the most computationally dense segments of the neural network to the cloud server, the adaptive system drastically reduces the total number of floating-point operations performed locally. The energy expended in transmitting the highly compressed intermediate feature maps was found to be substantially less than the energy required to compute the final, fully connected layers of the network natively on the mobile processor. Furthermore, by distributing the computational load, the adaptive system prevented the mobile device from reaching critical thermal thresholds. In comparative testing, devices forced to run pure edge inference experienced severe thermal throttling after consecutive evaluations, degrading subsequent inference times by a wide margin. The adaptive approach maintained stable device temperatures, ensuring consistent performance throughout the duration of the simulated clinical deployment.

4.4 Diagnostic Integrity and Robustness

A paramount concern in modifying medical diagnostic pipelines is the preservation of diagnostic integrity. Any mechanism introduced to improve speed or efficiency must not compromise the accuracy, sensitivity, or specificity of the clinical output. The adaptive inference stage partitioning framework proposed herein strictly preserves the exact mathematical operations of the original neural network. Because the system merely changes the physical location of the computation without altering the weights, architectures, or sequence of the layers, the theoretical accuracy remains entirely identical to a monolithic execution. However, the introduction of feature quantization and compression during the network transmission phase introduces a minor degree of noise into the intermediate feature representations. Extensive sensitivity analysis was conducted to quantify the impact of this intermediate quantization on the final classification accuracy. The results indicated that leveraging sixteen-bit floating-point quantization for transmission yielded absolutely no statistically significant deviation in cataract grading accuracy compared to the unquantized baseline. The deep neural network exhibited a robust capacity to absorb minor precision losses at intermediate stages, maintaining high diagnostic fidelity while benefiting from drastically reduced transmission payloads. This confirms that the pipeline enhancements provide substantial operational benefits without violating the strict clinical requirements of the screening application.

5. Conclusion

The deployment of sophisticated artificial intelligence for automated cataract screening in remote and resource-constrained environments represents a critical advancement in democratizing global ophthalmic healthcare. However, the practical realization of these systems is heavily impeded by the severe limitations of mobile hardware and the volatile nature of rural telecommunications infrastructure. This paper has comprehensively detailed a novel, adaptive inference stage partitioning framework designed specifically to overcome these deployment barriers. By treating the deep learning pipeline as a dynamically distributable asset, our methodology successfully balances the computational load between the mobile edge device and the remote cloud server. Through continuous, real-time profiling of available network bandwidth, device processing capacity, and battery status, the intelligent decision engine consistently identifies the optimal execution split point for every individual screening event. The extensive experimental analysis validates that this adaptive approach dramatically minimizes end-to-end diagnostic latency, averting the catastrophic delays associated with

transmitting raw medical imagery over degraded cellular networks. Furthermore, the framework significantly conserves mobile device energy and mitigates thermal throttling, extending the operational viability of point-of-care screening devices during intensive clinical shifts. While the current framework demonstrates exceptional efficacy, future research endeavors should focus on integrating multi-modal sensor data and exploring federated learning paradigms to continuously refine the predictive models of the decision engine without compromising patient data privacy. Ultimately, the adaptive inference stage partitioning methodology provides a highly robust, scalable, and clinically viable architectural foundation for the next generation of mobile medical diagnostics, ensuring that sight-saving artificial intelligence can be reliably delivered to the marginalized populations that require it most [27].

References

Yang, Z., Hu, D., Guo, Q., Zuo, L., & Ji, W. (2023). Visual E 2 C: AI-driven visual end-edge-cloud architecture for 6G in low-carbon smart cities. IEEE Wireless Communications, 30(3), 204-210.

Yang, Y., Ma, X., Li, C., Zheng, Z., Zhang, Q., Huang, G., ... & Zhao, Q. (2021). Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. Advances in Neural Information Processing Systems, 34, 10299-10312.

Fang, Z., Yang, G., Lyu, W., Hong, Z., Zhong, S., Zuo, W., ... & Zhang, D. (2025). Cellular Infrastructure Sharing for Network Robustness: A Citywide Empirical Study. IEEE Transactions on Mobile Computing.

Yang, Z., Ji, W., Guo, Q., & Wang, Z. (2023, October). Javp: Joint-aware video processing with edge-cloud collaboration for dnn inference. In Proceedings of the 31st ACM International Conference on Multimedia (pp. 9152-9160).

Wang, S., Zhang, X., Wang, P., & Li, X. (2026). Design of Magnetic Sensor Array-Based PUFs for IoT Security. IEEE Transactions on Instrumentation and Measurement, 75, 1-9.

Zhou, D. (2025, December). M-VP2: Microservice-Oriented Vulnerability Patch Planning-A Cost-Aware Approachusing Multi-Agent Reinforcement Learning. In 2025 5th International Conference on Computer, Internet of Things and Control Engineering (CITCE) (pp. 248-254). IEEE.

Yang, Z., Liang, B., & Ji, W. (2021). An intelligent end–edge–cloud architecture for visual IoT-assisted healthcare systems. IEEE internet of things journal, 8(23), 16779-16786.

Chen, Y., Zhao, C., & Gao, Y. (2025). Blockchain applications in health insurance: a review of applications, challenges, and prospects. Frontiers in Blockchain, 8, 1699290.

Chen, L., Jiang, X., Xiao, P., Liu, X., & Haardt, M. (2025). Structured tensor decomposition for fdd massive mimo downlink channel reconstruction. IEEE Transactions on Signal Processing, 73, 887-902.

Burrows, M. (2006). The Chubby lock service for loosely-coupled distributed systems. In Proceedings of OSDI.

Hu, Y., Yang, Z., Zhao, C., & Ji, W. (2026, May). Adaptive guidance semantically enhanced via multimodal llm for edge-cloud object detection. In ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 12962-12966). IEEE.

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2012). Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In Proceedings of NSDI.

Luo, H., Li, Q., Cheng, H., Li, W., Sun, W., Zhao, W., & Liu, Z. (2025). A²Tformer: Addressing temporal bias and non-stationarity in transformer-based IoT time series classification. IEEE Internet of Things Journal, 12(20), 42198–42213.

Zhao, Y., & Han, Z. (2025, December). Safe and Adaptive Cloud Healing: Verifying LLM-Generated Recovery Plans with a Neural-Symbolic World Model. In 2025 7th International Academic Exchange Conference on Science and Technology Innovation (IAECST) (pp. 89-94). IEEE.

Tian, Y., & Yang, Z. (2026, May). SAEC: scene-aware enhanced edge-cloud collaborative industrial vision inspection with multimodal LLM. In ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 19747-19751). IEEE.

Yang, W., Qin, Y., Yi, Z., & Yang, Y. (2019, May). Content-based hyperbolic routing and push mechanism in named data networking. In ICC 2019-2019 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE.

Yang, W., Qin, Y., Jiang, Z., & Chu, X. (2021, June). Traffic Management for Distributed Machine Learning in RDMA-enabled Data Center Networks. In ICC 2021-IEEE International Conference on Communications (pp. 1-6). IEEE.

Qu, Q., Lu, S., Shang, L., Jung, S., Liang, Q., & Pan, C. (2025, August). Fast and Energy-Efficient Analog Accelerator for Vision Transformer. In 2025 IEEE 68th International Midwest Symposium on Circuits and Systems (MWSCAS) (pp. 827-831). IEEE.

Wuyang Zhang and Shichao Pei. 2026. Your LLM Agent Can Leak Your Data: Data Exfiltration via Backdoored Tool Use. arXiv:2604.05432 [cs.CR] doi:10.48550/arXiv.2604.05432

Yang, Z., Ji, W., Guo, Q., Zhao, J., Zhao, C., Zhang, X., ... & You, Y. (2026). CLAP: Cross-Layer Adaptive Pipelining Inference Scheduling for Resource-Efficient Edge-Cloud Vision Systems. ACM Transactions on Architecture and Code Optimization.

Yang, Z., Ji, W., & Wang, Z. (2024). Adaptive joint configuration optimization for collaborative inference in edge-cloud systems. Science China Information Sciences, 67(4), 149103.

Shen, B., Wang, Y., Shang, D., & Ning, T. (2026). Self-Commitment Latency: A Reward-Free Probe for Prompted Implicit Hacking. Publication date: 2026-06-04.

Sun, L., Wang, W., Yuan, T., Mi, L., Dai, H., Liu, Y., & Fu, X. (2024, May). BiSwift: Bandwidth orchestrator for multi-stream video analytics on edge. In IEEE INFOCOM 2024-IEEE Conference on Computer Communications (pp. 1181-1190). IEEE.

Li, X., Wang, P., Li, G., Ni, L., & Zhang, Y. (2023). Design of interface circuits and lightweight PUF for TMR sensors. IEEE Sensors Journal, 23(11), 11754-11761.

Yang, W., Qin, Y., & Li, R. (2022). A network-embedding-based approach for scalable network navigability in content-centric social IoT. IEEE Internet of Things Journal, 9(17), 16418-16428.

Zhou, D. (2026). AI-Driven Hybrid SAST–DAST–SCA–IAST Framework for Risk-Based Vulnerability Prioritization in Microservice Architectures.

Li, Q., Li, L., Liu, Z., Sun, W., Li, W., Li, J., & Zhao, W. (2025). Cloud-edge collaboration for industrial Internet of Things: Scalable neurocomputing and rolling-horizon optimization. IEEE Internet of Things Journal, 12(12), 19929–19943.