International Journal of Basic and Applied Sciences, 14 (3) (2025) 332-337

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/cbjvn374
Research paper

A Comparative Study of Computing Paradigms for Real-Time Image Processing: Cloud, Edge, And On-Device AI

Won-hyuk Choi [1] *, Woo-Jin Jung [2]

[1] Department of Avionics, Hanseo University
[2] Department of Aeronautical System Engineering, Hanseo University
**Corresponding author E-mail: choiwh@hanseo.ac.kr*

Received: June 16, 2025, Accepted: June 23, 2025, Published: July 27, 2025

Abstract

With the growing demand for real-time image and video processing, selecting an efficient computing architecture has become increasingly important. This study conducts a comparative analysis of three paradigms—cloud computing (CC), edge computing (EC), and on-device edge computing (ODEC)—to determine the most suitable method for real-time applications. By evaluating their performance across varying data volumes and network conditions, we identify the trade-offs in latency, scalability, and responsiveness. The findings reveal that cloud computing suffers from increased latency due to transmission bottlenecks, while edge and on-device computing significantly reduce latency through decentralized processing. Among the three, ODEC demonstrates the most consistent performance, particularly in environments requiring large-scale data handling and minimal network dependence. These results suggest that on-device AI offers a promising direction for future real-time systems by addressing the limitations of traditional architectures.

Keywords: *Cloud Computing; Edge Computing; AI-Based On-Device Edge Computing; Real-Time Processing; Data Analysis.*

1. Introduction

In recent years, the explosive growth of digital data has underscored the critical importance of efficient data processing and management strategies. As a result, modern data infrastructures increasingly rely on cloud computing, edge computing, and lightweight AI-powered edge computing to meet ever-expanding requirements. These three paradigms have emerged as core solutions for handling diverse data-driven applications. Cloud computing remains a standard approach for managing extensive datasets and optimizing resource allocation, especially within enterprise environments and large-scale online platforms. More recently, the integration of artificial intelligence into cloud services—often termed "Cloud AI"—has enabled organizations to accelerate and refine their data analysis capabilities, leading to improved business decision-making. On the other hand, edge computing addresses the challenge of network congestion by processing data closer to its source. By analyzing and filtering significant volumes of data locally—rather than offloading everything to remote cloud servers—edge computing enables rapid response and minimizes latency, which is especially advantageous for time-sensitive applications utilizing IoT devices. Building upon these concepts, lightweight AI edge computing brings advanced data analytics directly to edge devices, enabling real-time processing and immediate action even in resource-constrained environments. This approach is particularly beneficial for scenarios requiring instant results, such as emergency response, location-based services, and real-time medical monitoring. This paper investigates and compares the underlying algorithmic models and hardware implementations of these three approaches, focusing on their effectiveness in demanding, high-volume data processing scenarios, such as continuous monitoring of corridors or security zones. By systematically evaluating their respective strengths, weaknesses, and performance boundaries, we aim to clarify the optimal use cases for each technology and provide practical guidance for their deployment in different operational contexts. 1Faculty of Social Sciences and Humanities, Putera Batam University, Indonesia. E-mail: mortigor.afrizal@gmail.com

2. Computing data processing research

2.1. Edge computing

Edge Computing (EC) is a technology that processes large-scale data generated in an IoT environment at the boundary of a network to speed up data analysis and provide rapid services to users. In particular, the importance of edge computing is remarkable in situations such as corridor monitoring, which requires large amounts of data to be processed in real time. Corridor monitoring is one of the appropriate cases for the use of edge computing because it collects and analyzes large amounts of data in real time. When edge computing is applied, data collected from the corridor monitoring device can be processed immediately on the spot and quickly provide the information necessary for real-time decision-making. This is effective in reducing delays that can occur by directly processing data collected from IoT devices on the spot, unlike cloud computing methods that process all data on a central server. In addition, the edge cloud can minimize service latency and efficiently utilize the resources of the central server by initially collecting and analyzing IoT data on the spot and then transmitting only the necessary information to the central cloud. This method can optimize the performance of IoT services and greatly improve the efficiency of data processing.

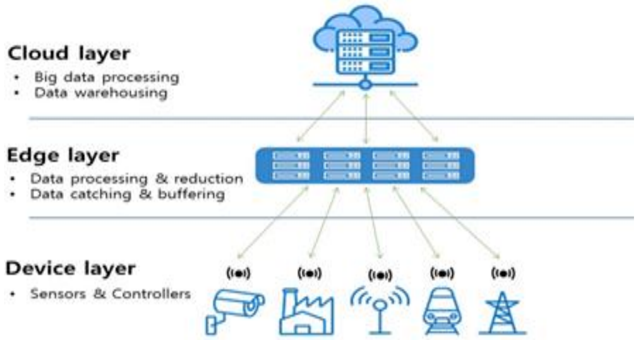

Fig. 1: Data Flow Structure in Cloud-Edge-Device Paradigm.

2.2. Cloud computing

Cloud Computing (CC) provides better processing performance than edge devices, and the conversion of computing tasks to the cloud is widely recognized to increase the efficiency of data processing. Cloud computing performs well in processing large amounts of data through a centralized structure and provides important advantages in situations where large-scale data, such as corridor monitoring, must be processed in real-time. This has the potential to increase the efficiency of enterprise IT resource management and improve data processing speed. Cloud computing is utilized not only in data storage and processing, but also in various industries and plays an important role in responding to continuous technological development and market demand. These technologies provide the possibility to fundamentally change the IT infrastructure of companies and individuals, and their importance is gradually emerging. However, the centralized structure of cloud computing can cause bottlenecks in the process of large-scale data transmission, which can cause problems such as service delays or interruptions. To solve this problem, this paper compares the utility of on-device AI computing (On-device AI Computing) technology. On-device AI minimizes delays and reduces bottlenecks that may occur during the data transmission process by processing in real time on the device itself without transferring data to the cloud. This technology can greatly improve stability, security, and performance, and provides a way to build a cooperative computing environment between the cloud and edge devices while complementing the shortcomings of cloud computing. This cooperative approach enables faster and more stable data processing while maintaining the scalability of cloud computing.

2.3. On-device AI edge computing

On-Device AI-based intelligent edge computing (OEC)** is a technology that processes, analyzes, and transmits data by executing an AI model on the device itself, and has the characteristics of providing real-time responsiveness and large-scale data processing capabilities at the same time. This is a very advantageous technology in situations where large amounts of data have to be processed in real-time, such as real-time video processing. Corridor monitoring is a task of monitoring and predicting the marine environment in real-time, and is a representative example in which intelligent edge computing using on-device AI can be effectively applied. On-device AI has the advantage of being able to respond in real time without transferring data to an external server by processing data directly inside the device.

This method is suitable for rapid collection and analysis of large-scale data occurring in the ocean and provides the ability to process immediately at the device level. Compared to cloud computing, which often suffers from high latency and increased energy consumption due to data transmission and centralized processing, on-device AI significantly reduces end-to-end latency and enhances energy efficiency by eliminating the need for continuous communication with remote servers. Local devices are responsible for real-time processing and initial analysis, and data centers support higher-order tasks such as more complex analysis and large-scale data storage. Cloud services complement complex operations that devices cannot handle and increase the efficiency of the entire system. By combining virtualization and artificial intelligence (AI) technologies, on-device AI can convert vast amounts of data collected from devices into valuable information in real time. AI algorithms can detect complex or unexpected phenomena more accurately and have the ability to analyze them in real-time. These characteristics allow on-device AI-based intelligent edge computing to greatly improve the accuracy and efficiency of real-time image processing tasks, and to quickly respond to emergency situations such as disasters by utilizing mobility and local processing power. This study analyzes the technical characteristics and performance of on-device AI-based intelligent edge computing to explore the development potential of smart systems that can overcome the limitations of cloud computing and traditional edge computing. The on-device AI-based intelligent edge computing architecture used for real-time processing is presented in Figure 2.

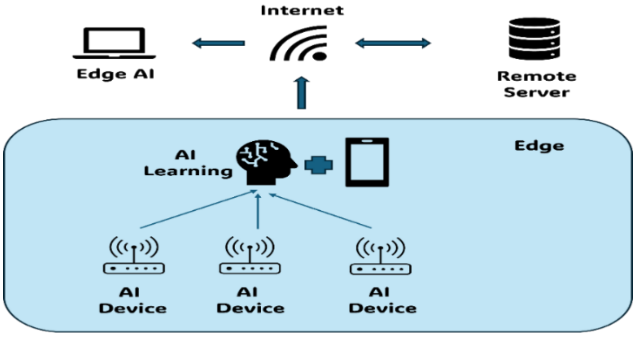

Fig. 2: AI-Based Intelligent Edge Computing Architecture.

2.4. Mathematical modeling of local delay in computing architectures

The following is an equation for deriving the parameters used for the experiment

2.4.1. Delay calculating parameter

Parameters to calculate the local delay in computing methods.

$$\mathrm{D_q} = \text{ Queuing Delay (Delays caused by server queuing, according to methods)} \tag{1}$$

$$\mathrm{D_d} = \text{device Delay (Delays caused by server hardware spec, according to methods)} \tag{2}$$

$$\mathrm{D_b} = \frac{\text{Data size(byte)}}{\text{Bandwidth}} \text{(Delays caused by getting data according to methods)} \tag{3}$$

$$\mathrm{D_p} = \frac{\text{Distance}}{\text{Propagation Speed}} \text{(Delays caused by propagation of signals through media, according to methods)} \tag{4}$$

$$\mathrm{D_n} = \mathrm{D_q} + \mathrm{D_b} + \mathrm{D_d} \text{ (Additional delays caused by the bottleneck section, according to methods)} \tag{5}$$

2.4.2. Cloud computing

Total local delay formula for cloud computing.

$$\mathrm{D_c} = \mathrm{D_p} + \mathrm{D_b} + \mathrm{D_d} + \mathrm{D_q} \text{ (total delays in cloud computing)} \tag{6}$$

2.4.3. Edge computing

Total local delay formula for edge computing.

$$\mathrm{D_e} = \mathrm{D_b} + \mathrm{D_d} \text{ (total delays in Edge computing)} \tag{7}$$

2.4.4. On-device AI edge computing

Total local delay formula for On-device AI edge computing.

$$\mathrm{D_e} = \mathrm{D_b} + \mathrm{D_d} \text{ (total delays in AI edge)} \tag{8}$$

2.4.5. Infra penalty

Infra penalty is a parameter that combines bottlenecks and latency factors, such as data compression rates inherent in cloud computing and existing edge computing, into a single factor when AI edge computing is set as the default value.

$$\mathrm{P_{Cloud}} = (\text{node count} \times \mathrm{D_n}) + (1 - \text{Data compression\%p}) \tag{9}$$

$$\mathrm{P_{edge}} = (1 - \text{Data compression \%p}) \tag{10}$$

3. Research results

3.1. Temperature simulation results

In this study, various environments for the implementation of cloud computing, edge computing, and on-device edge computing were set and compared. In Table 1, parameter values are set so that fair comparisons can be made between different computing models. In this way, you can see how effectively each computing model works in different environments and requirements. In the case of On-device AI Edge, the 30% compression rate that has undergone average data processing through AI was set as a parameter to differentiate it.

Table 1: Parameter Table

Parameters	Setting
Data Size, R	10000 MB
Bandwidth Cloud, B_c	800 Mbps
Bandwidth Edge, B_e	80 Mbps
Bandwidth AI Edge, B_a	80 Mbps
GPU Cloud, G_c	8000 GFLOPS
GPU Edge,AI Edge, G_e, G_a	1000 GFLOPS
CPU Cloud, C_c	1500 GFLOPS
CPU Edge,AI Edge,C_e, C_a	100 GFLOPS
RAM Cloud, F_c	256GB
RAM Edge,AI Edge, F_e, F_a	16GB
Distance Factor Cloud, d_c	0.08
Distance Factor Edge, AId_e, d_a	0.016

compression AI Edge, c_a	30%
Infra penalty	0, 100, 200

3.2. Compute processing performance comparison

Figure 3 illustrates the average processing time across increasing data sizes, providing a clear comparison between cloud computing, edge computing, and on-device AI-based intelligent edge computing. As shown in the graph, cloud computing maintains a relatively flat processing time curve even as data size increases. This consistency stems from the cloud's centralized infrastructure, which leverages high-performance servers and stable computational resources that are less affected by input data volume.

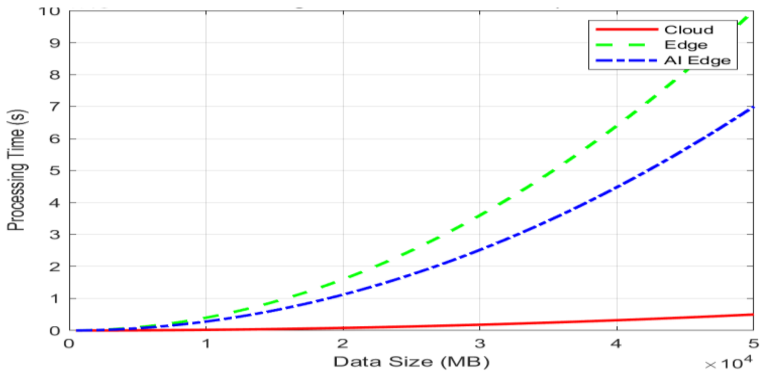

Fig. 3: Average Processing Time by Data Size.

In contrast, both edge computing and on-device AI-based edge computing exhibit a more noticeable increase in processing time as data size grows. This trend reflects the influence of limited local resources and bandwidth when data is processed closer to the source. However, on-device AI outperforms standard edge computing across all data sizes. This performance gain is attributed to the AI system's ability to process data directly on the device without relying heavily on external computation or transmission. Notably, on-device AI-based edge computing proves particularly effective in environments with limited or unstable network connectivity. Since data is processed locally, system performance is not significantly affected by network delays, making it well-suited for applications requiring real-time responsiveness. Additionally, the ability to scale processing through distributed nodes ensures that even large volumes of data can be handled efficiently without incurring the steep latency increases typically seen in centralized systems.

3.3. Compute processing performance comparison with penalty added

Figure 4 illustrates the locally observed processing latency as a function of node count, incorporating additional infrastructure factors and penalties such as transmission distance, network bottlenecks, and node distribution. As the number of nodes increases, the average delay in cloud computing systems rises steeply. This is primarily due to the centralization of data processing and the accumulation of network bottlenecks as all data is transferred to a centralized server, which severely impacts scalability. In contrast, edge computing and on-device AI-based edge architectures demonstrate significantly lower and more gradual increases in latency, even as the node count grows. This is attributed to their ability to process data locally at the edge, thereby minimizing the volume of data that must be transmitted over the network and preventing the formation of bottlenecks. The intelligent, distributed nature of on-device AI systems allows for real-time processing at each node, ensuring that increases in latency remain linear and manageable even with larger deployments.

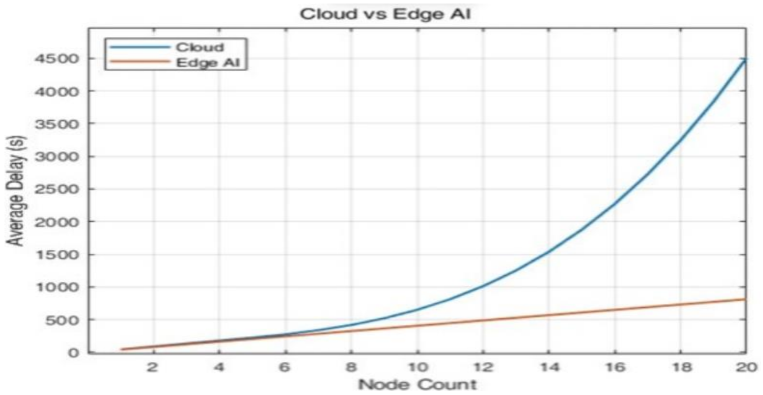

Fig. 4: Average Latency Comparison between Cloud and Edge AI Architectures.

As depicted in Figure 4, in this experimental environment, there is a gradual average delay in processing time from the 6th node to the 12th node. If the number of nodes increases by more than 20, it is possible to see a bottleneck in which the average processing time explodes. Therefore, edge-based computing technology showed a low increase in terms of multi-node.

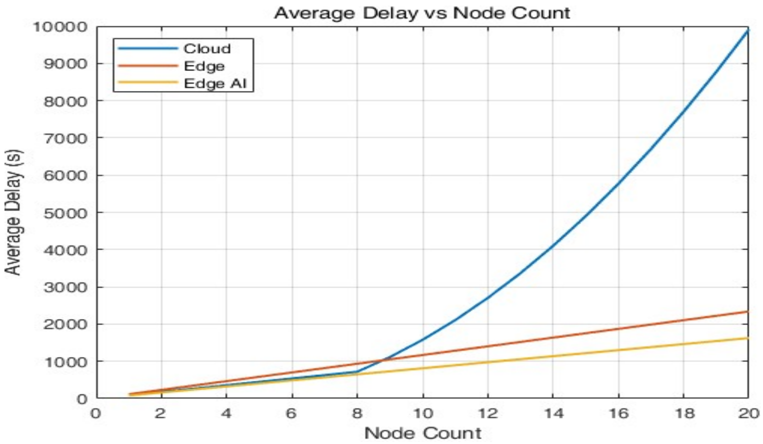

Fig. 5: Average Delay as A Function of Node Count for Cloud, Edge, and Edge AI Architectures.

Figure 5 provides a comprehensive comparison of average processing delay as the number of nodes increases, evaluated across three computing paradigms: cloud, edge, and edge AI. The results demonstrate that as node count rises, the cloud architecture exhibits an exponential increase in average delay. This is primarily due to the cumulative effect of centralized data aggregation and processing, which amplifies bottlenecks and transmission overhead as more nodes simultaneously transmit data to the central server. Edge computing alleviates some of this burden by distributing processing tasks closer to the data source. As a result, edge-based systems display a noticeably slower growth in latency compared to traditional cloud architectures. However, the delay still increases with the number of nodes due to residual aggregation and network limitations at the edge layer. Most notably, edge AI systems—which leverage intelligent, on-device data processing and localized decision making—consistently achieve the lowest average delay across all node counts. By minimizing both the volume of data transmitted and the reliance on centralized infrastructure, edge AI not only prevents network bottlenecks but also ensures that processing latency grows linearly, even as node density scales. As depicted in Figure 6, the advantage of edge AI becomes especially pronounced as deployments reach higher node counts, reinforcing its suitability for scalable, real-time applications in distributed environments.

4. Conclusions and future research directions

The experimental results presented across Figures 4, 5, and 6 demonstrate the unique strengths and limitations of cloud computing, edge computing, and on-device AI-based intelligent edge computing. Each paradigm offers distinct advantages depending on the application context, network environment, and real-time processing requirements. Cloud computing exhibits stable processing times even with increasing data sizes, owing to powerful centralized infrastructure. It is well-suited for large-scale data analytics and batch processing. However, its performance is heavily dependent on network bandwidth and suffers from high latency, making it less ideal for real-time applications. Edge computing improves upon this by processing data closer to the source, significantly reducing latency. As seen in the node-based delay analysis, edge computing scales better than cloud computing in distributed environments. Nevertheless, its performance still degrades with increasing data complexity and volume due to constrained local resources. On-device AI-based intelligent edge computing consistently demonstrates the lowest average delay and processing time under various workloads and node configurations. By minimizing data transmission and enabling real-time processing directly at the device level, it outperforms traditional edge solutions, particularly in environments with unstable or constrained network conditions. Additionally, local data processing inherently enhances system security and reduces exposure to centralized threats. Despite these advantages, on-device AI solutions face limitations in computational power compared to cloud infrastructures. However, these challenges can be mitigated through advanced techniques such as efficient model compression, hardware acceleration, and the evolution of lightweight AI architectures optimized for edge devices. In real-time critical environments such as corridor monitoring or autonomous driving, it is essential to design systems that selectively refine and prioritize data within the computational boundaries of edge devices. This kind of system-level trade-off design—balancing latency, accuracy, and energy efficiency—plays a crucial role in ensuring responsiveness while respecting hardware constraints at the edge. In conclusion, the choice between cloud, edge, and intelligent edge computing should be guided by specific system requirements: cloud computing for large-scale batch operations, edge computing for latency-sensitive distributed environments, and on-device AI for ultra-low-latency applications with limited network dependency. Future research should focus on hybrid computing models that dynamically allocate workloads across cloud and edge environments, and on enhancing the efficiency of on-device AI through software optimization and specialized hardware advancements.

Acknowledgements

Research was conducted through the Hanseo University on-campus research support project.

References

[1] Fernandes, V.; Carvalho, G.; Pereira, V.; Bernardino, J. Analyzing Data Reduction Techniques: An Experimental Perspective. Appl. Sci. 2024, 14, 3436. https://doi.org/10.3390/app14083436.

[2] Siddiqui, S.T.; Khan, M.R.; Khan, Z.; Rana, N.; Khan, H.; Alam, M.I. Significance of Internet-of-Things Edge and Fog Computingin Edu-cation Sector. In Proceedings of the 2023 International Conference on Smart Computing and Application (ICSCA), Hail,Saudi Arabia, 5–6 February 2023; IEEE: Piscataway, NJ, USA; pp. 1–6. https://doi.org/10.1109/ICSCA57840.2023.10087582.

[3] Lee, H.; Kim, J.; Park, S. Edge AI vs. Cloud AI: A Comparative Study of Performance, Latency, and Scalability. Appl. Sci. 2025, 15, 1289.

[4] Nasir Abbas at al "Mobile Edge Computing: A Survey" IEEE internet of Things Journal, Vol. 5, No. 1, pp. 450-465, Sep 2017. https://doi.org/10.1109/JIOT.2017.2750180.

[5] Y. Mao et al., "A Survey on Mobile Edge Computing: The Communication Perspective," IEEE Commun. Surveys Tuts, Vol. 19, no. 4, pp. 2322–2358, Apr. 2017. https://doi.org/10.1109/COMST.2017.2745201.

[6] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” Future Generation Computer Systems, Vol. 29, No. 7, pp.1645-1660, Sept. 2013. https://doi.org/10.1016/j.future.2013.01.010.

[7] Chen, A.; Liu, F.H.; Wang, S.D.e. Data reduction for real-time bridge vibration data on edge. In Proceedings of the 2019 IEEE Internation-al Conference on Data Science and Advanced Analytics, DSAA, Washington, DC, USA, 5–8 October 2019; pp. 602–603. https://doi.org/10.1109/DSAA.2019.00077.

[8] Kyung-rae. C ; Seok-min.H; Won-hyuk.C . Performance Comparison and Optimal Selection of Computing Techniques for Corridor Sur-veillance Journal of the Korean Society of Navigation, 2023, 27.6: 771-776.

[9] IFTIKHAR, Sundas, et al. AI-based fog and edge computing: A systematic review, taxonomy and future directions. Internet of Things, 2023, 21: 100674. https://doi.org/10.1016/j.iot.2022.100674.

[10] HUA, Haochen, et al. Edge computing with artificial intelligence: A machine learning perspective. ACM Computing Surveys, 2023, 55.9: 1-35. https://doi.org/10.1145/3555802.

[11] JMERENDA, Massimo; PORCARO, Carlo; IERO, Demetrio. Edge machine learning for ai-enabled iot devices: A review. Sensors, 2020, 20.9: 2533. https://doi.org/10.3390/s20092533.