

Data Optimization Using Edge AI: A Framework for Efficient Real-Time Analytics - A Case Study of IndoAI AI Camera

Vivek Gujar

Founder-Director, IndoAI Technologies P Ltd

Email: [vivek\[at\]indo.ai](mailto:vivek[at]indo.ai)

Abstract: *In the era of rapidly expanding smart surveillance systems, AI cameras are pivotal for real-time analytics in applications ranging from security to industrial automation. However, with the increasing number of connected cameras, there is an immense need for efficient data management, especially concerning memory usage, data transmission and cloud dependency. This article explores the integration of Edge Technology in AI surveillance systems, specifically focusing on the optimization of data generated by AI cameras. IndoAI's AI cameras, equipped with real-time facial recognition, fire detection and vehicle number plate detection, produce large volumes of data that necessitate a scalable, efficient solution. This paper explores a data optimization framework using Edge AI, where data is processed closer to its source- on the AI camera itself - thereby minimizing transmission and reducing reliance on cloud services. The framework leverages edge processing techniques, intelligent compression algorithms and inter-camera communication to reduce redundant data, optimize real-time analytics and ensure efficient network performance. The proposed system is inspired by data optimization techniques used in OTT streaming platforms, particularly in adaptive bitrate streaming, which dynamically adjusts content quality based on bandwidth availability.*

Keywords: AI Camera, IndoAI, Data Analytics, Edge AI, Appization, Custom AI Models

1. Introduction

The demand for real-time data analysis in AI cameras has grown significantly in recent years, driven by the widespread adoption of smart cities, autonomous systems, and advanced security infrastructure. As per S&P Global Smart cities could become even smarter with increased application of AI, both in infrastructure development and analysis of data [1]. However, the traditional cloud-centric model of data processing is becoming less feasible for large-scale deployment due to bandwidth limitations, latency issues[2], and high cloud storage costs. Extremely high network bandwidth and low latency requirements would place unprecedented pressures on traditional cloud-based AI, where massive sensors/embedded devices transfer collected data to the cloud, often under varying network qualities (e.g., the bandwidth and latency). To address these problems, edge AI is the answer [6] Edge AI—where data processing occurs locally on the device—emerges as a solution. By optimizing how AI cameras handle and transmit data, Edge AI can drastically reduce the volume of information sent to the cloud, decrease latency, and enable faster real-time decision-making. This paper outlines the key components of a data optimization framework built on Edge AI, offering solutions to the primary challenges of scalability, redundancy and efficiency. Edge AI,

or "AI on the edge"[3], refers to the combination of edge computing and artificial intelligence to execute machine learning tasks directly on interconnected edge devices. Edge computing allows for data to be stored close to the device location, and AI algorithms enable the data to be processed right on the network edge, close to where the data is located [4], rather than centrally in a cloud computing facility or private data center with or without an internet connection. This facilitates the processing of data within milliseconds, providing real-time feedback. Edge AI refers to the deployment of artificial intelligence (AI) algorithms and models directly on edge devices, such as embedded systems, smartphones, Internet of Things (IoT) devices, and other local computing devices, rather than relying on cloud-based computing resources [5]. Edge AI pushes inference and training processes of AI models to the network edge in close proximity to data sources [6].

An edge-based network is a network located at the edge of a centralized network that brings data storage and computation as near the required point as possible, pushing applications, data, and computing power away from the centralized data center in order to deliver low latency and save bandwidth, as shown in below fig1 [7]. The edge infrastructure is the same as the cloud infrastructure, but it does its tasks at the edge of the network.

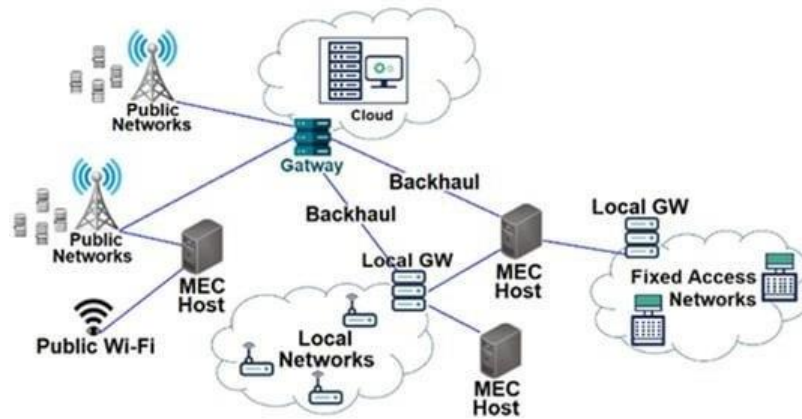


Figure 1

Differences between Edge Computing and Edge AI based on scope, objective, computing infrastructure and functionality [8]:

Aspect	Edge Computing	Edge AI
Scope and Objective	Distributed computing paradigm that brings resources closer to the data source to reduce latency and optimize network bandwidth. It focuses on real-time data processing at the edge of the network.	Deployment of AI algorithms/models directly on edge devices. The focus is on enabling AI capabilities for local data processing, real-time decision-making, and intelligent behavior at the edge.
Computing Infrastructure	Involves edge servers, gateways, or routers at the network edge. These devices provide computational power and storage for data processing closer to the source.	Leverages the computing infrastructure of edge devices like smartphones, IoT devices, and embedded systems. AI models are deployed directly on these devices for local AI processing.
Functionality	Optimizes data flow, reduces latency, and improves application performance by processing data closer to the edge. It performs tasks like data aggregation, filtering, and forwarding to the cloud.	Builds on edge computing by adding AI capabilities. Enables intelligent data processing and decision-making at the edge. Edge AI algorithms perform tasks such as object recognition, predictive analytics, and real-time responses directly on the device.

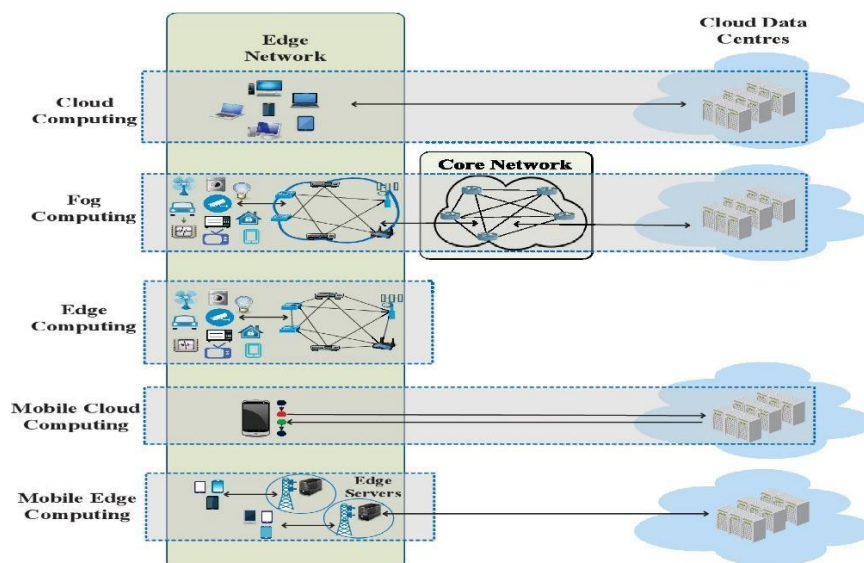


Figure 2

Above fig 2 presents a comparative figure that provides a quick insight into the differences [9]

2. Edge AI and Data Optimization: A Conceptual Framework

Edge AI, several techniques are key to optimizing performance and resource use in distributed environments:

Dynamic Model Selection

By dynamically adapting to the contextual needs of each inference job, edge devices can maximize their overall efficiency, providing accurate results while respecting strict time and energy limitations. Rather than relying on a single model, Edge AI can deploy multiple models with different trade-offs between accuracy and speed [21]. The system dynamically switches models based on conditions like weather. For instance, a simpler object detection model may suffice in clear weather, while a more complex one is necessary during rain [10]. A meta-model, possibly using

reinforcement learning, performs a **cost-benefit analysis** to decide the optimal model.

Distributed Inference

It makes devices connect seamlessly and process data in real time, delivering instant insights by bringing advanced computing power closer to the edge, reducing latency, and enhancing efficiency [22].

In networks with many edge devices, like CCTV cameras, built-in redundancy allows systems to tolerate failures or less accurate results. Distributed inference algorithms leverage global data, enabling the system to recover from errors by balancing high- and low-accuracy models. This ensures resource optimization while maintaining overall accuracy.

Online Training and Refinement

Training at the edge or potentially among “end–edge–cloud,” treating the edge as the core architecture of training, is called “AI Training at Edge” [24]. Some edge applications need models to continue learning post-deployment. Edge devices can send data samples back to a central cloud model, which refines the local models. This process, similar to **knowledge distillation**, allows the edge models to adapt and improve in real time [10].

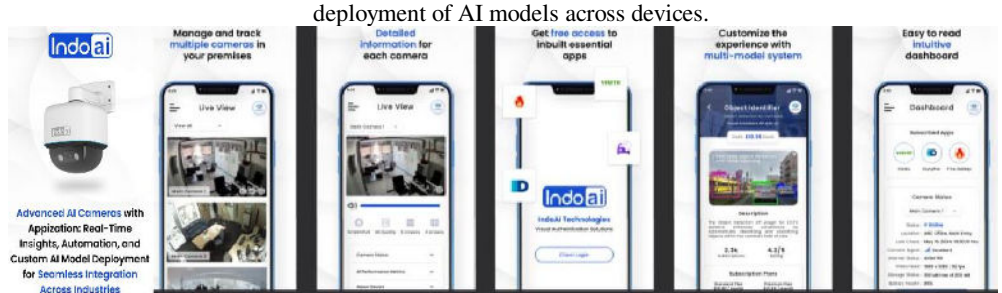
Federated Learning

With large-scale edge deployments, **federated learning** enables edge devices to train local models and send only model weights to the cloud. This preserves bandwidth, enhances privacy, and reduces communication costs. Federated learning [23] enables the creation of ML models without compromising user data privacy. Users train individual models and periodically share updates with a central server. The server aggregates these models, creating a centralized model that incorporates insights from all users without accessing their specific data

These approaches collectively make Edge AI more efficient, scalable, and responsive, positioning it as a critical technology for real-time applications across industries.

3. Case Study: IndoAI AI Camera [26] & its Framework

A walkthrough of feature & description of IndoAI AI Camera is presented below in a tabular format.

Feature	Description
Edge AI Computing Concept	Data is processed locally on the AI camera, reducing the need to transmit large volumes of raw data to the cloud. Edge analytics reduce power consumption through data reduction, prediction, and transformation, enhancing real-time decision-making.
Minimized Cloud Dependency	The camera handles most data processing locally, sending only critical or summarized data to the cloud, reducing network congestion and cloud dependency.
Edge Processing Techniques	AI cameras perform tasks like facial recognition, fire detection, and vehicle detection on-device, reducing raw data transmission by sending only processed data or significant anomalies for further analysis.
Intelligent Compression	Optimizes data sent to the cloud by using compression algorithms that maintain data integrity while reducing size, ensuring efficient bandwidth use and lower storage costs.
Inter-Camera Communication	AI cameras communicate with each other to share relevant information and avoid redundant data collection, optimizing the monitoring area and minimizing duplicated data transmission.
Redundant Data Elimination	Edge processing and inter-camera communication eliminate redundant data by using deduplication techniques, ensuring only unique, necessary data is transmitted or stored.
Optimized Real-Time Analytics	Local processing enables real-time analytics like fire detection or vehicle recognition without relying on cloud-based processing, providing immediate actionable insights.
Efficient Network Performance	Edge processing reduces data load on the network, transmitting only essential data, ensuring stable network performance and intelligent use of bandwidth.
Scalability	The system is scalable, allowing more AI cameras to be added without significantly increasing bandwidth or cloud costs, benefiting from edge computing and inter-camera optimization.
Energy Efficiency	Local processing reduces cloud communication and energy consumption required for data transmission, improving overall energy efficiency.
Security and Privacy	Sensitive data is processed locally, enhancing privacy and security by reducing transmission risks and potential breaches during data transfers.
Customizable AI Models	Custom AI models can be downloaded onto the camera for specific tasks like facial recognition or fire detection, further enhancing local data processing and optimizing overall camera performance.
AI Model Synchronization via Appization	One unique feature of IndoAI’s system is the concept of “Appization,”[26] where different AI models (e.g., facial recognition, fire detection) can be dynamically downloaded and applied to AI cameras. This system is built using containerization technologies such as Docker and Kubernetes, which allow for scalable deployment of AI models across devices. 

4. Framework

The proposed data optimization framework for AI cameras integrates Edge AI processing, intelligent compression, and inter-camera communication to optimize memory use, reduce transmission bandwidth, and improve real-time analytics.

4.1 Components of the Framework

1) Edge AI Processing

Edge artificial intelligence refers to the deployment of AI algorithms and AI models directly on local edge devices such as sensors or Internet of Things (IoT) devices, which enables real-time data processing and analysis without constant reliance on cloud infrastructure.[11]

Edge AI, which refers to the local processing of AI algorithms on edge devices, it negates the need to transmit data to the cloud for processing, enabling real-time decision-making and substantially reducing communication costs associated with cloudlet platforms. In essence, edge AI brings processing and computational tasks closer to the point of interaction with the end-user, whether that be a smartphone, single board computer (SBC), domestic appliance, IoT device, or edge server [12].

AI cameras are equipped with onboard processing units, such as NVIDIA Jetson Nano / Google Edge TPU, capable of handling computationally intensive tasks like facial recognition, fire detection, and vehicle number plate recognition directly on the device.

- Each camera processes raw data locally, identifying and classifying events of interest. This ensures that only meaningful data is uploaded to the cloud, thereby reducing bandwidth consumption.
- For example, in a fire detection scenario, only the frames where smoke or fire is detected are sent to the cloud, rather than the entire video stream.

2) Intelligent Compression Algorithms

In data reduction, the data are compressed because sending compressed data requires less power compared to sending fewer data. In data prediction, instead of data, a data model is maintained to predict the future data [13].

AI models for edge devices [14] are often adapted using **model compression**, which reduces their size and complexity without losing performance. Key techniques include:

- Pruning: Removing unnecessary connections and neurons from the model.
- Quantization: Lowering the precision of the model's weights and activations.
- Knowledge Distillation: Training a smaller model to imitate a larger, more accurate one.

Another challenge is choosing the right algorithms that can efficiently run on hardware with limited resources. Common algorithms used in Edge AI include:

- CNNs (Convolutional Neural Networks) for image recognition.
- RNNs (Recurrent Neural Networks) for processing sequences of data.
- Reinforcement Learning for decision-making tasks.

These algorithms can be optimized for edge devices by utilizing hardware-specific features like parallel processing and AI accelerators.

IndoAI AI Camera

- *Inspired by adaptive bitrate streaming used in OTT platforms like Netflix, intelligent compression algorithms are proposed-implemented on AI cameras to reduce the size of the data being transmitted to the cloud.*
- *H.265 (HEVC) video compression codec is used to reduce file sizes while maintaining video quality. The camera detects and sends only the critical parts of the footage (e.g., a face or vehicle in motion), compressing or even discarding irrelevant frames.*
- *Additionally, object-level compression is employed, where only objects of interest (e.g., a detected vehicle or person) are transmitted at high resolution, while the rest of the frame is compressed to a lower resolution or discarded.*

3) Inter-Camera Communication

Shi et al [6] opines for full potential, the upcoming edge AI, shall rely on advances in various aspects, including the smart design of distributed learning algorithms and system architectures, supported by efficient communication protocols. In the proposed system of Alsmirat et al [15], each cell consists of multiple video sources, such as cameras or video sensors, all sharing a common communication medium. These video sources can be mobile and battery-powered. Each cell is equipped with an edge server, co-located with the cell's access point, ensuring a high-bandwidth connection that eliminates potential bottlenecks. The edge server processes computer vision algorithms and can send automated alerts when suspicious activities or objects are detected within the cell. In their case study, face detection serves as the primary computer vision task. The framework enables the video sources to capture and transmit data to the edge server, which then performs intelligent, cross-layer optimization of both the hardware resources of the video sources and the network bandwidth. The network can be wireless, allowing for mobile video sources, further enhancing system flexibility.

IndoAI AI Camera

- *Multiple cameras in a network collaborate by sharing data with one another to avoid redundant processing. If one camera identifies a vehicle, it communicates this information to nearby cameras, so they do not need to reprocess the same vehicle.*
- *This allows for cooperative edge processing, where a group of cameras collectively processes data and shares insights, significantly reducing redundant computations and optimizing network traffic.*
- *5G connectivity facilitates fast communication between cameras, enabling quick synchronization of data across multiple nodes in the system.*

4) Real-Time Analytics and Decision-Making

According to Sukhpal et al [16] Edge AI plays a critical role by enabling real-time insights and actions without the need to transmit vast amounts of data to centralized cloud systems. By localizing data processing, Edge AI reduces cloud dependency, minimizes latency, and enhances the responsiveness of applications to real-time inputs. This approach is essential for advancing next-generation technologies that require immediate data analysis and

decision-making, positioning Edge AI to revolutionize data-driven decision-making processes.

IndoAI AI Camera

- *By processing data locally, AI cameras provide real-time alerts for critical events like fire detection or unauthorized access, allowing for immediate responses.*
- *The system ensures low-latency decision-making in mission-critical applications, such as fire alarms in industrial plants or security breaches in smart cities.*

Cloud Offloading and Storage Optimization

- *Not all data needs to be sent to the cloud. Events of interest are stored locally and sent to the cloud only when needed for long-term storage, further analysis, or when the local memory reaches capacity.*
- *The cloud acts as a secondary layer for storage and advanced analytics, while routine tasks are managed at the edge.*
- *For scenarios requiring cloud access, AWS Greengrass or Microsoft Azure IoT Edge frameworks can be used to ensure smooth integration between edge and cloud systems.*

5) Software and Technologies Used

The optimization framework relies on a combination of hardware and software technologies to implement Edge AI effectively. Although optimization approaches tackling this problem from either an algorithmic or a hardware perspective exist, hardware-software co-design methodologies are key [17]. With the right combination of hardware and software, Edge AI technology can enable AI algorithms to run on edge devices, bringing intelligence to the network's edge and enabling new applications and use cases [7]

Key technologies and software include:

- TensorFlow Lite and PyTorch Mobile [14,18]: *TensorFlow Lite is a robust and versatile platform designed for deploying machine learning models on Edge devices. Lightweight versions of deep learning frameworks that enable AI models to run on resource-constrained devices. PyTorch and TensorFlow are the most popular deep learning frameworks designed to process a lot of data.*
- OpenCV: An open-source computer vision library used for processing video streams and identifying objects in real time.
- YOLO (You Only Look Once): *YOLO stands out for its speed and efficiency, making real-time object detection a reality [19]. The YOLO algorithm represents a significant advancement in the field of object detection, particularly in edge AI applications [20]. An AI model used for real-time object detection, enabling facial recognition, vehicle detection and fire detection with high accuracy.*
- AWS IoT Greengrass: A framework that extends cloud functionality to edge devices, allowing AI cameras to perform local computing and communication between edge and cloud systems.

6) Scalability and Network Efficiency

As the number of connected AI cameras increases, network efficiency becomes paramount. The proposed system incorporates several features to ensure scalability:

- **Distributed Edge Processing:** As cameras process data locally, the load on the central cloud is minimized, allowing more cameras to be added to the system without overwhelming network resources.
- **Hierarchical Data Processing:** Cameras are organized in a tiered network, where edge cameras send data to intermediary processing nodes (e.g., local servers), which further aggregate and compress data before sending it to the cloud.
- **Self-Learning AI Models:** Over time, AI cameras adapt to their environment, reducing false positives and improving detection accuracy. These self-learning models ensure that only critical data is processed, further optimizing bandwidth and memory usage.

5. Conclusion

This article proposes a robust data optimization framework using Edge AI, designed to address the growing challenges of memory management, bandwidth optimization, and cloud dependency in AI camera networks. By integrating edge processing, intelligent compression and inter-camera communication, IndoAI's cameras will be able to deliver real-time analytics with greater efficiency, scalability and accuracy.

As the deployment of AI cameras scales across industries, this Edge AI-based optimization framework will ensure seamless, scalable performance, enabling rapid growth in sectors such as smart cities, security, industrial automation, and more. According to EPOSS [25] Edge AI is revolutionizing the data analytics landscape by reducing cloud dependency while key challenges include:

- Developing new algorithms and neuromorphic-based chips,
- Integrating specialized computing platforms with traditional systems,
- Enhancing automated transfer learning for federated learning,
- Optimizing neural networks for specific applications,
- Creating tools for semi-automatic design and deep network generation,
- Building open architecture for faster deployment (open-source SW, data, platforms, HW), and
- Implementing energy-efficient AI training while ensuring security, privacy, and explainability.

Future research will explore the integration of emerging technologies like 6G and blockchain for further enhancement of network security and data integrity.

References

- [1] <https://www.spglobal.com/en/research-insights/special-reports/ai-smart-cities>
- [2] Saurabh Shukla, Mohd. Fadzil Hassan, Duc Chung Tran, Rehan Akbar, Irving Vitra Papatungan, Muhammad Khalid Khan, 2023, Improving latency in Internet-of-Things and cloud computing for real-time data transmission: a systematic literature review (SLR), Cluster Computing (2023) 26:2657–2680, <https://doi.org/10.1007/s10586-021-03279-3>
- [3] <https://ibm.com/topics/edge-ai>

