kinara

OPTIMIZING LATENCY FOR EDGE AI DEPLOYMENTS

DESIGNING THE OPTIMAL SYSTEM FOR LOW LATENCY VIDEO ANALYTICS APPLICATIONS

TECHNICAL ANALYSIS

Introduction

With more than a billion surveillance cameras installed worldwide, along with the rapid growth of camera-enabled devices, video has become the most pervasive form of data generated. It is imperative for organizations to optimize the processing and analysis of this video data to extract valuable insights and subsequently make more informed decisions in real-time. Video analytics is the process of using advanced algorithms and deep learning techniques to analyze video footage in real-time or near real-time to extract useful information and insights.

Video analytics has become increasingly important in many industries, including security, retail, transportation, and manufacturing, as it enables businesses to automate the process of monitoring video feeds and identify events or objects of interest. By leveraging the power of analytics, organizations can enhance safety, improve operational efficiencies, and provide better customer experiences, ultimately gaining a competitive edge in their industry.

To be effective at video analytics, a system must be designed to optimize the following parameters:

Latency is the most critical metric for real time video analytics applications

- **Latency**: The latency of the system is a critical metric. Latency is the time it takes to respond and process video data and provide information. This is especially important for the majority of edge AI applications that must run multiple models for each video frame. The system must have low latency to ensure real-time analysis and decision-making capability. This contrasts with a system that is designed only for high throughput, running the same task and AI model continuously on a video stream.

- **Scalability**: The system's scalability is an important capability to accommodate growing needs, changing requirements, and evolving technologies.

- **Accuracy:** The accuracy of the video analytics system in detecting and analyzing events is also important for confidently processing relevant events and avoiding false alarms.

- **Robustness**: The ability of the system to operate in a wide range of environmental or deployment conditions is an important consideration.

In this white paper, we'll explore the importance of latency in video analytics and how it impacts various industries. We'll also discuss strategies for optimizing latency in edge AI deployments, allowing businesses to achieve lightning-fast response times and make more informed decisions based on real-time data.

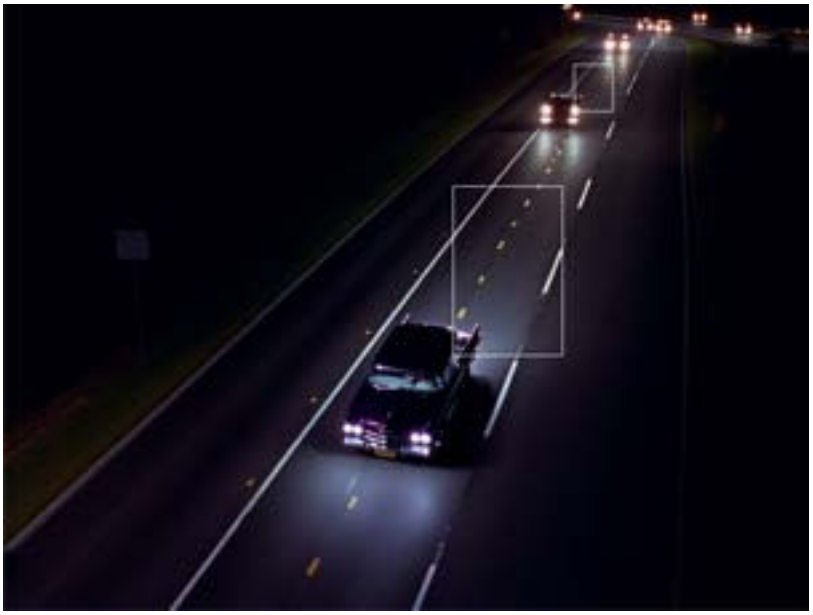


Figure 1. High latency causes bounding boxes to be incorrectly drawn

Latency considerations for real life video analytics

Video analytics applications can be classified in two categories in terms of their latency requirements:

a. **Strict latency requirement**: For mission-critical applications, the system must provide results within a specific maximum time, beyond which the results are considered outdated, irrelevant, or inaccurate (Figure 1).

b. **Soft latency requirement**: These applications don't result in an immediate system failure if there is no response within a specific period. However, not meeting the target time affects other aspects like usability and may increase deployment cost (e.g., necessitating local and/or cloud storage).

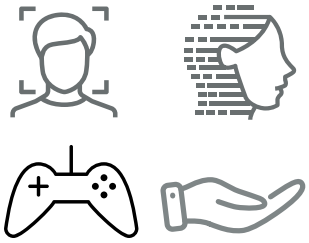
For mission critical applications, high latency can lead to a critical system failure

Here are some examples of video analytics applications for each type of latency category:

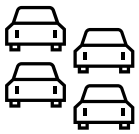
Strict Latency Requirement

- **Gesture or motion-controlled video gaming**: In motion-controlled computer gaming, a gamer's physical movements are captured and translated into corresponding in-game actions. Low latency is critical for providing a smooth and responsive gaming experience.

- **Cashierless stores**: Cashierless stores using computer vision for automated checkout require a strict latency of 33ms or less because the system must track people in real-time as they move in the store, take objects from shelves, and automatically pay for them. Note: 33ms references the standard camera frame rate of 30 frames per second, therefore, a latency greater than 33ms will result in dropped frames.



- **Intelligent traffic systems**: Video analytics for intelligent traffic systems involves using advanced computer vision techniques to analyze data from cameras placed at key roadway locations and toll booths. One example is license plate recognition, allowing authorities to track vehicles and more effectively enforce traffic laws. As cameras capture the license plates for every passing car, the system should detect the license plate number in real-time to reduce both bandwidth and storage costs (as opposed to transmitting the video stream to the cloud for processing).



Soft Latency Requirement

- **Retail analytics**: Video analytics is increasingly used in retail applications to understand customer behavior, optimize inventory, and prevent loss. While most of these applications don't have a strict latency requirement, the captured videos add to the storage cost if data is not processed on the edge and in real-time. Especially for loss prevention (i.e., reducing shrinkage), it is important to identify any event as early as possible.

- **Industry 4.0**: These applications involve AI for manufacturing automation and often have a low latency requirement for safety. Low latency allows for real-time monitoring and control of industrial processes, enabling operators to identify and address issues as they arise. For a machine vision system that inspects products on an assembly line for defects, low latency ensures that any identified defects are detected and addressed immediately.

Building a low latency system

We can now explore how to build a system that is optimized for low latency. In a typical edge environment, IoT devices capture the data at a certain rate (for example, a video camera can capture videos at 30 fps). The actual computation can be either in the IoT device, on the cloud, or a hybrid of the two. For video analytics specifically, the users can run the AI applications inside a smart camera, on an edge server, or in the cloud (Figure 2).

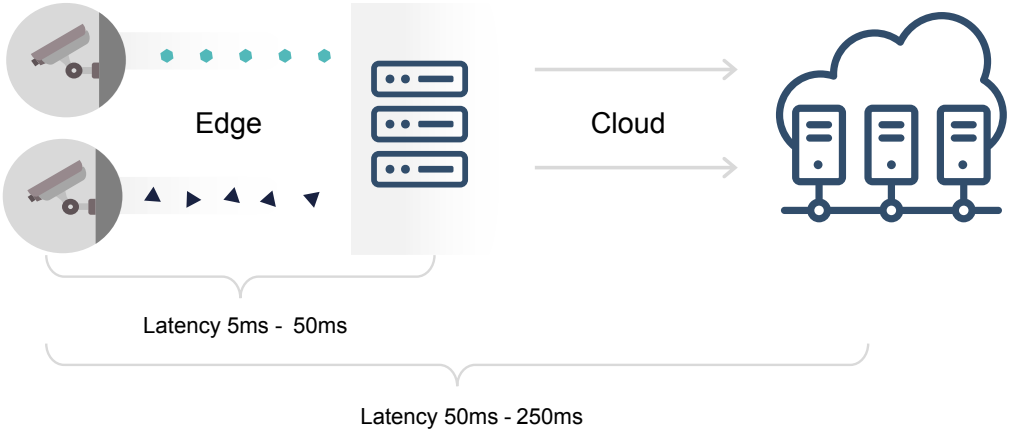


Figure 2. High level architecture of a video analytics system depicting the latency discrepancy between edge and cloud.

Video analytics in the cloud will have a higher latency than on the edge due to the need to transfer video data. Cloud-based video analytics can achieve a minimum latency of around 50 milliseconds. However, latencies of less than ten milliseconds may be required for many applications requiring near-real-time processing, such as robotics. Furthermore, with sufficient processing capabilities, many models can be run directly in the camera to achieve even lower latency than edge servers.

A hybrid approach can be used to reduce latency where most AI processing is done on edge devices, while only sending metadata to the cloud for further analysis. This approach can help achieve the latency threshold while still leveraging the scalability and flexibility of cloud.

The most optimal way to reduce latency is by running inference at the edge

Optimizing latency for the edge

The overall latency in the edge environment is a combined function of the following:

a. Capture latency: Capture latency refers to the time it takes for a camera or sensor to capture the video frame and transfer it to the processing system. This includes the time taken to digitize the video signal and the delay in the camera's image sensor and image processing pipeline.

b. Network latency: Network latency is the delay introduced during the transmission of video data between the cameras and the computing device or server. This latency can be influenced by factors such as the available bandwidth, network congestion, and the physical distance between devices. Note that this time is virtually zero if the capture and compute happen within the same device.

c. Compute latency: Compute latency for AI refers to the time it takes to execute the AI pipeline and the analytics application once the video data is available. This includes tasks such as decoding the video, running analytics algorithms (including inference), and encoding the results. The application might also require the execution of multiple models per video frame, strongly emphasizing the need for being able to switch models with zero overhead.

To minimize latency, each system component must be optimized by carefully selecting hardware, software, and network configurations that meet the system's unique requirements. Both capture and network latency, which are a function of the infrastructure, have improved dramatically in recent years. The capture speed is rapidly increasing due to advances in sensor technology. Even relatively inexpensive cameras can record 60fps videos with 4k resolution and above. Network communication has also improved significantly – including adoption of 5G-mmwave, Wi-Fi 7, and Ethernet switches that prioritize video content.

Kinara Ara-1

However, compute latency is still a bottleneck, highly dependent on the complexity of the video analytics algorithms and the processing capabilities of the edge computing device or server. The most effective way to reduce the compute latency is by using a dedicated, latency-optimized AI processor.

For deep neural networks specifically, models have a very high degree of parallelism and data reuse. Reducing latency for neural network inference requires a combination of hardware and software techniques, including quantization and hardware acceleration.

Kinara Ara-1 edge AI processor is architected for low latency AI processing.

Approaches to reduce compute latency

While system developers have traditionally used GPUs to leverage its parallelization and compute density for AI inference, GPUs are not inherently designed for low latency required for many edge applications. In fact, many modern GPUs have prebuilt tensor cores or separate deep learning accelerators (DLA) to run AI inference more efficiently, although these features add to the GPU complexity, cost, and memory usage. Alternatively, System-on-chip (SoCs) for IoT devices can include integrated AI accelerators. These integrated AI accelerators will always be limited by the system-level overhead of the associated SoC. A cost-effective and more efficient approach for low latency edge inference can be accomplished by adding a standalone accelerator (e.g., Kinara's Ara-1) to an SoC.

Ara-1's patented 'polymorphic dataflow architecture' minimizes data movement across the chip and between compute and various levels of the memory hierarchy. Furthermore, this architecture supports zero switching time when executing multiple models. Ara-1 also utilizes an offline scheduler that minimizes the model's weights and parameter data movement when running the inference. Neural networks have different data flow across different models and even across layers within a given network. The architecture of Ara-1 leverages this network property to optimize latency across a range of AI models.

Ara-1, which is manufactured on a TSMC 28nm process, delivers much lower latency than other AI accelerators (e.g., Intel's MyriadX and Google's Edge TPU manufactured on a 16nm and 28nm process, respectively) due to its architectural superiority (Table 1 and Table 2).

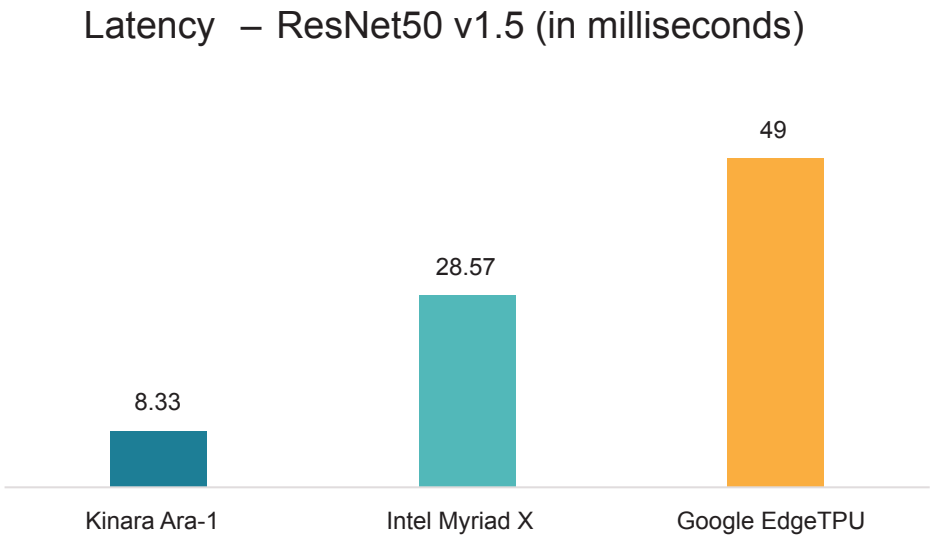


Table 1. This table compares the latency (msec) when running ResNet 50 v1 with a batch = 1. Note: smaller is better.

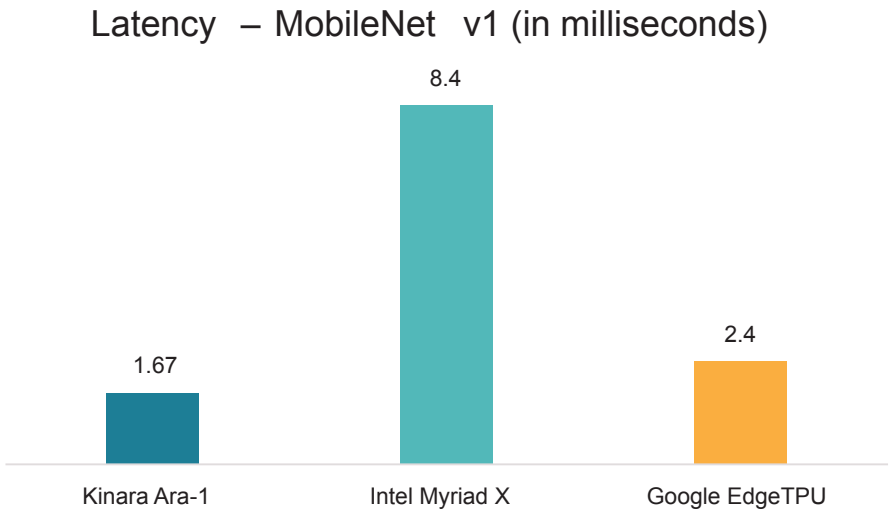


Table 2. This table compares the latency (msec) when running MobileNet v1 SSD with a batch = 1. Note: smaller is better .

Kinara's advantages include the following:

- Minimize costly data movement to reduce latency and power: Resource planner leverages the d[1]ataflow cores to implement the execution plan with maximum reuse
- Low-overhead task graph management: HW implementation allows for full control over chip execution flow while minimizing latency
- Enables users to run multiple models with zero-switching time on the same video frames.
- Software-first approach: Kinara compiler optimizes and orchestrates the end-to-end execution of the neural network
- Algorithmic flexibility: Dataflow ISA cores efficiently implement any current and future neural network operators

1. Intel Myriad numbers from https://docs.openvino.ai/2020.1/_docs_performance_benchmarks.html
2. Google edge tpu numbers from: https://coral.ai/docs/edgetpu/benchmarks/

Conclusion

Optimizing latency for edge AI deployments is critical for analyzing the ever-increasing volume of video data generated by surveillance cameras and camera-enabled devices.

The most effective way to reduce compute latency is by utilizing dedicated, latency-optimized AI processors, such as Kinara's Ara-1 processor. With its polymorphic dataflow architecture and software-first approach, Ara-1 minimizes data movement across the chip and within the memory hierarchy, leading to reduced latency and power consumption. Additionally, Ara-1's architecture supports zero switching time between models, which is particularly beneficial for applications that require running multiple models consecutively. For edge AI deployments, latency is a critical benchmark, and optimizing latency can be significantly improved using dedicated latency-optimized AI processors like Kinara's Ara-1.

By embracing such innovative technologies and approaches, businesses can unlock the full potential of edge AI deployments, enhance user experiences, and gain a competitive advantage in their respective industries.