kinara

BUILD SMARTER CAMERAS AND EDGE AI APPLIANCES WITH KINARA ARA-1 PROCESSORS AND NXP i.MX APPLICATIONS PROCESSORS

TECHNICAL ANALYSIS

The abundance of high-resolution cameras can enable many use cases by applying AI to the video streams. But that requires AI solutions at the edge that perform neural-network inferencing on streaming video at varying frame rates and with varying application complexity. With their combinations of CPUs, GPUs, and even neural accelerator, NXP® Semiconductors' i.MX applications processors provide AI capability for many levels of performance. However, there are also a growing number of edge AI applications in retail, health monitoring, security, industrial control, and others, that require complex, higher accuracy models, the ability to handle video at higher frame rates (30fps or more), resolutions at 1080p and beyond, and with multiple models running in parallel. By combining NXP's i.MX applications processors with Kinara's Ara-1 Edge AI processors, we can build a low-cost, low-power system delivering the very high level of AI compute required by these demanding use cases.

The most common deployment scenarios for AI applications are smart cameras and edge-AI appliances. In this paper, we'll describe how NXP's i.MX applications processors can be used in these scenarios combined with Kinara's Ara-1.

Building Smart Cameras With i.MX 8M SoCs and Ara-1

For a smart camera, the vision processing and support for neural-network inferencing are integrated into the main camera board. The camera performs complex AI tasks before sending the application's metadata to an edge server or the cloud for non-real-time analytics.

When combined with the Ara-1 processor, host processors like the i.MX 8M Nano and i.MX 8M Plus SoCs are extremely well suited for a low-power smart-camera design. Fig 1(a) shows one such design with an image sensor, i.MX 8M Plus applications processor and one or more Ara-1 chips integrated on a single camera board. Since the i.MX 8M Plus SoC includes dual image signal processors it can interface directly to a camera sensor and convert raw sensor data to the RGB format needed for display, storage, and inferencing. In the case of other NXP SoCs such as the i.MX 8M Nano SoC, which don't include an ISP, an image sensor with an integrated ISP can be used as part of the smart camera design.

AI Inference Flow for Smart Camera

The host processor acquires the video frames from the sensor, and performs pre-processing steps to transform this sensor data into a format matching the input requirements of the AI inferencing performed on the Ara-1 (Figure 1b). Specifically, normalization, mean-subtraction, scale-type activities (e.g., resize, crop), quantization to int8 data format, are performed on the host processor, typically using either the CPU cores or the GPU. The quantized input is then sent to Ara-1 for inference over USB or PCIe interface. Following the inference, there are post-processing functions that must be run. An example of post processing is non-maximal suppression (NMS) logic used to interpret the output of most object detection models. Post processing is typically performed on the host processor, though in some cases it can also be offloaded to Ara-1. In either case, the post-processed results are then available to the application to implement the desired business logic.

Smart Camera

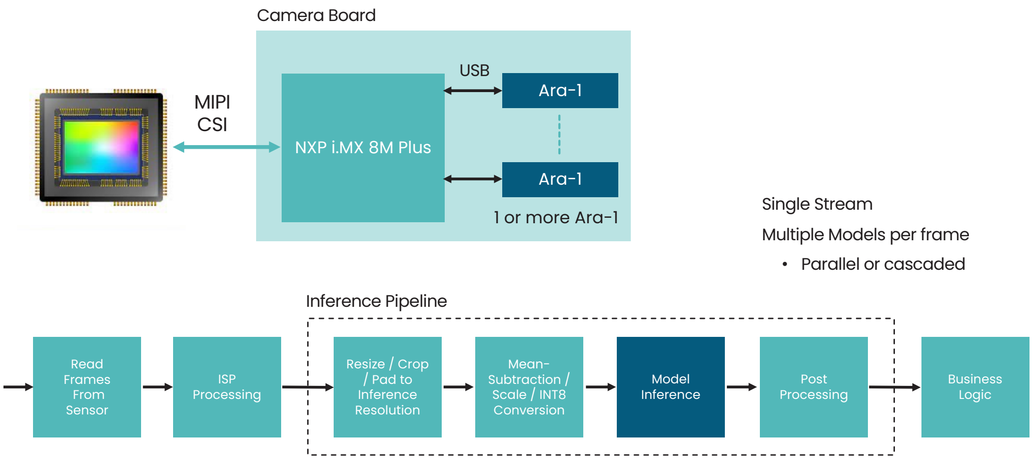


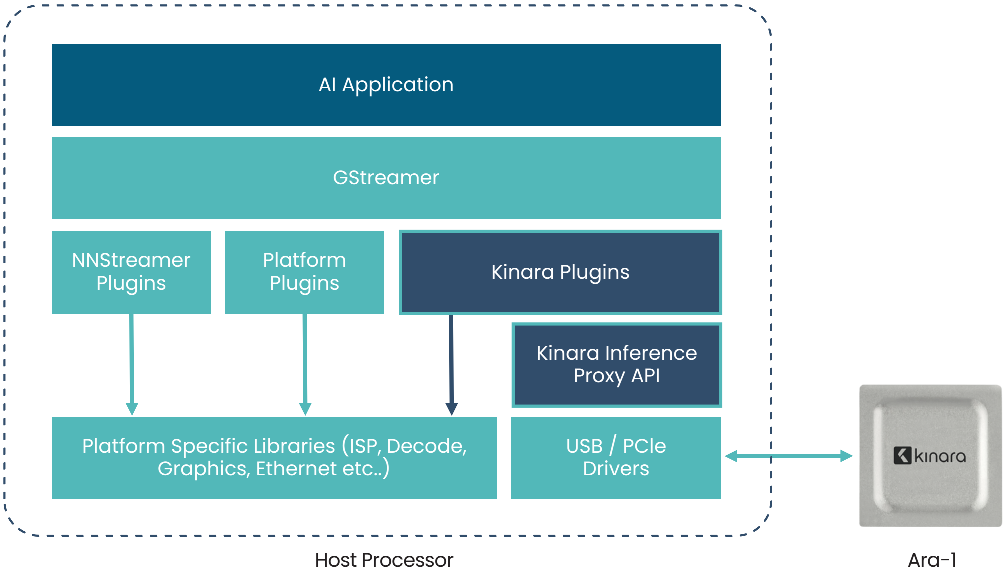
Figure 1.
Smart Camera application flow

GStreamer Based AI Pipelines

An AI application must manage the execution of these pre-and-post processing steps as well as the inferencing on Ara-1. Moreover, most smart camera applications require multiple models to be executed per frame or per stream. These could be either parallel inference pipelines with multiple models running independently of each other, or a cascaded pipeline where one model learns, then based on the result of that model, another one or more instances of another model is executed.

NXP platforms provide GStreamer and NNStreamer frameworks to simplify deployment of such AI pipelines. Regardless of which i.MX SoC you deploy and whether you run the pre-processing on CPU or GPU, GStreamer is used as a framework for creating streaming media applications, abstracting the hardware layer to allow the use of any i.MX SoC without having to change the underlying vision pipeline software. As a library designed for the construction of compute graphs of media-handling components, GStreamer makes it easy to assemble pipelines of filters to handle any multi-model scenarios.

NNStreamer plugins allow Gstreamer developers to efficiently adopt neural network models into the vision pipeline that runs on the CPUs, GPUs, or NPU core present in the i.MX applications processors. Kinara has developed a set of Gstreamer compatible plugins, that make it seamless to integrate Ara-1 into NXP inference pipelines.



Building Edge AI Appliances With i.MX 8M Applications Processor and Ara-1

An alternative to building smart cameras is using an edge AI appliance for AI computations. In this case, the cameras can be regular IP cameras without any AI capabilities - feeds from multiple cameras and other sensors are routed to an AI appliance for inferencing. Typically, each AI appliance handles 4-16 camera and sensor feeds. In a large facility with hundreds of cameras, several such appliances are deployed. This enables a much more scalable processing model compared to edge servers where all the feeds from these hundreds of cameras need to be routed to a set of central servers, incurring substantial infrastructure costs and expensive, power-hungry central servers. **(For more on this, check our white paper "Building the Ideal Checkout Free Store")**

Unlike the direct sensor interface used in a smart camera, an Edge AI appliance typically receives video feeds over an IP network using ethernet or Wi-Fi. Moreover, the feeds are generally encoded and the appliance must perform video decode on each incoming stream before it can be processed. Finally, unlike a smart camera, an appliance must perform inferencing and associated pre-and-post processing tasks on multiple streams. Such an appliance would include multiple Ara-1 processors for higher AI compute capability as well as a more powerful host processor.

One such design uses NXP's i.MX 8M Quad applications processor as the host (Figure 2). Its hardware-accelerated, video decode support allows real-time decode of up to eight 1080p streams or two 4K streams. Furthermore, the i.MX 8M's powerful quad core CPUs and GPU can handle the pre- and post-processing functions of multiple inference pipelines.

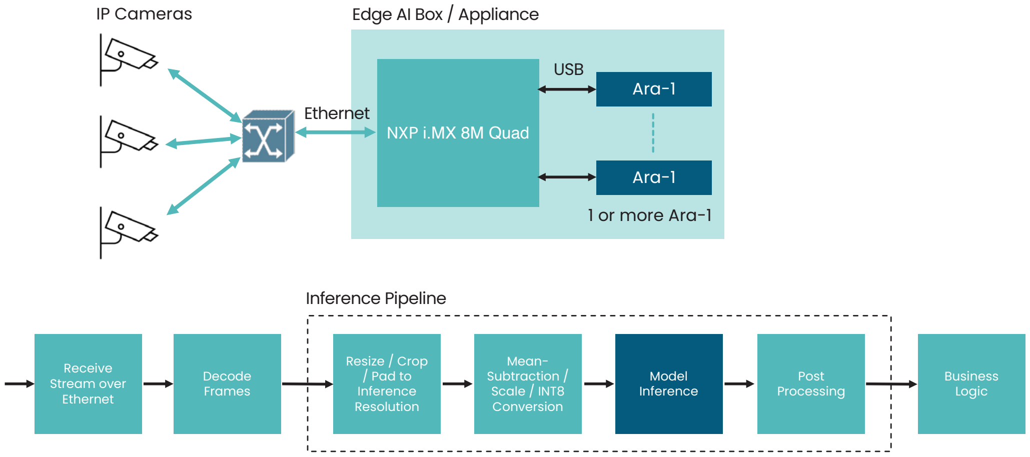


Figure 2.
Edge AI Appliance application flow

Choosing the Right Number of Ara-1 Devices

The maximum number of streams supported by the edge appliance is partly a function of the host processor and how many streams it can decode and perform pre- and post-processing on. Similarly, the number of Ara-1 devices in the appliance determines how many AI processing streams can be supported. A single Ara-1 device can handle multiple video streams, however, the number of streams per device is tied to model complexity and size. For example, an eight stream AI edge appliance typically requires anywhere from 1 to 4 Ara-1 devices, depending on the model complexity and desired processing rate.

Like the smart camera use case, GStreamer can again be used to handle data acquisition, pre and post processing and inferencing for multiple streams and multiple models. In a multi-device configuration, the Kinara runtime driver can perform automatic load balancing across the multiple Ara-1 devices on a per inference basis. This helps ensure uniform work distribution across the Kinara devices. Or you could set it up at the application layer to tie specific streams to specific Ara-1s.

Building Flexible AI Solutions

Whether building smart cameras or edge AI appliances, a developer can integrate inferencing functions into a wide range of NXP i.MX applications processors. As discussed above, different i.MX SoCs serve different functions; for example, with the i.MX 8M Plus and i.MX 8M Quad SoCs with its decode support for eight 1080p streams or two 4K streams. From a hardware design perspective, combining any i.MX SoC with Ara-1 is simplified because of the common PCIe or USB support. As we've seen, from a software perspective, Kinara utilized GStreamer to easily add Ara-1 into the inference pipeline and eliminate major changes to code when switching i.MX applications processors.

KINARA | LEADING EDGE AI

Kinara is deeply committed to designing and building the world's most power- and price-efficient edge AI inference platform supported by comprehensive AI software development tools. Designed to enable smart applications across retail, medical, industry 4.0, automotive, smart cities, and much more; Kinara's AI processors, modules and software can be found at the heart of the AI industry's most exciting and influential innovations. Led by Silicon Valley veterans and a world class development team in India, Kinara envisions a world of exceptional customer experiences, better manufacturing efficiency and greater safety for all. Kinara is a member of the NXP Partner Program. Learn more at **www.kinara.ai**