VISION INSIGHTS

VISION SYSTEMS DESIGN®

PHOTO ID 146955410 © FUNTAP P | DREAMSTIME.COM

A Primer: Developing Embedded Vision Applications

What is the proper mix between processing and analyzing image information at the edge versus in the cloud?

Embedded vision is typically defined as image capture and processing—and often involves analysis with intelligent vision algorithms—running on the same piece of hardware on the so-called edge, or where the action is happening.

Some applications may divvy up the processing between an edge device and a local area network or the cloud. Another option is to send only important results—such as products that fail inspection—to the cloud for archival storage.

In this collection of articles, we explore the types of machine vision applications suited to an embedded, or edge computing, approach; the role of FPGAs in embedded applications; sensor requirements; and how to deploy edge learning, or AI-enabled applications running at the edge.

Whether you work at an OEM, system integrator or end-user organization, we've provided insights here to help fuel your next project.

Linda Wilson,
EDITOR IN CHIEF
VISION SYSTEMS DESIGN

Vision Processing: At the Edge or in the Cloud? "BLERP."

PHIL LAPSLEY

It can be a tricky question, one with multiple answers—some right, some wrong! And, it's often challenging to think through. At the Edge AI and Vision Alliance, we use the funny acronym BLERP (bandwidth, latency, economics, reliability, and privacy) to help decide where processing should take place. Following are five of the most important factors to consider when making edge-cloud tradeoffs.

Bandwidth

Obviously, vision processing in the cloud requires a network connection over which to send your images. Depending on your application, your bandwidth requirements could be a trickle (say you're sending one small image of a dumpster to determine whether it's full or not) or a flood (say you're monitoring hundreds or even thousands of cameras in real time at a grocery store). Whether you can get a network connection that will accommodate your requirements is another story. In some cases, you may not have the option for an Internet connection at all (e.g., a wildlife camera out in the middle of nowhere), and in others, you might have a reasonably reliable, low-cost pipe to the cloud (think of a consumer's WiFi-connected doorbell). Balancing your requirements

and available network capacity is critical. Edge systems excel when bandwidth needs are high or available bandwidth is low (or non-existent).

Latency

Some applications require instant answers. Think about self-driving cars, for example. If your cloud-based image processing system takes several seconds to recognize an object, you might have already run over a pedestrian and be halfway down the block by the time you realize you should have stopped. But other systems can tolerate much longer latency. A camera based system to recognize what food items you have in a refrigerator, for example, might be happy taking tens of seconds, or even minutes, to realize you've put a carton of milk in the fridge. The lower your latency requirements, the more pressure to do your processing at the edge.

Economics

The best things in life may be free, but, sadly, bandwidth and computing aren't among them. If you've ever paid for cloud computing, you know how quickly those costs can add up. Similarly, just look at your cable or cell phone bill to get an idea of how expensive bandwidth can be. Edge processing can save you money in bandwidth charges (the more you do on device, the less you do in the cloud), but adding a more capable processor to your product costs money. For many products, a key business insight is understanding who is paying for compute and bandwidth, what their willingness to pay is, and whether they're already paying for them in some form. For example, if you're making a consumer appliance that uses a homeowner's existing Internet, well, they're already paying for that network connection; from your perspective, bandwidth is free. Conversely, if your customer is willing to pay for a more capable device (or better still already has such a device, like maybe a high-end mobile phone), that can save you money by reducing your cloud computing costs by offloading processing to the edge.

Reliability

Is it important that your system continue to function if there's a network outage? A facial-recognition-based home door lock, for example, probably needs local processing (at least as a fallback!) if the homeowner's WiFi network goes down. In general, the more critical reliability is, the greater the need for edge processing.

Privacy

To paraphrase the Vegas slogan, "What happens at the edge stays at the edge." Stated differently: If you're not sending images or video up to the cloud for processing, there's nothing in the cloud for the bad guys to steal, nor for you to accidentally leak via a misconfigured AWS S3 bucket. Not only does this reduce your liability, it's also a great selling point for your customers that care about privacy.

Where you do your vision processing is a balancing act and depends on a variety of factors: bandwidth, latency, economics, reliability, and privacy chief among them. Thinking through how these five factors relate to your application can help determine whether processing should take place at the edge, in the cloud, or some combination of the two. The ultimate answer depends on the specific requirements and constraints of your particular application, of course, but we hope that a BLERP analysis can help you reach an answer that's right for you.

Phil Lapsley is a co-founder of embedded-AI consulting firm BDTI and a vice president of the Edge AI and Vision Alliance, a 100+ member industry association dedicated to inspiring and empowering innovators to create systems that perceive and understand. He is also an organizer of the Embedded Vision Summit, an annual event.

How FPGAs Are Used in Embedded Vision Applications

Offering a combination of low power, advanced computation, and security, FPGAs suit applications ranging from artificial intelligence to drones.

APURVA PERI

With the emergence of high-resolution, bandwidth-hungry surveillance equipment, artificial intelligence integration, real-time analytics, and the rapid rise in Internet of Things (IoT) adoption, there's been increased focus on edge computing for video data processing workloads. While edge computing offers such benefits as improved response time, optimized bandwidth budgets, and data privacy, it also poses complex challenges such as data and design security, and restricted power and footprint budgets. These constraints demand the need for specialized computing hardware with advanced security features.

Mid-range field programmable gate arrays (FPGA) can be defined as those with a logic density count in the range of 100 to 500 K as compared to higher density FPGAs with logic densities ranging from 1 to 9 M. High-range FPGAs are popularly used in applications such as advanced driver assistance systems and data centers, while mid-range FPGAs deploy into a variety of embedded applications such as surveillance, gateway devices, small cell wireless applications, medical and industrial imaging, and unmanned aerial vehicle (UAV) monitoring.

The Roles of FGPAs in Technology Evolution

Edge computing is distributed computing that, compared to centralized computing, can improve response time and conserve bandwidth by bringing computation and data storage closer to the application. With the ever-burgeoning IoT, an increasing number of commercial and industrial applications rely on the connected web of sensor-based, digital, and mechanical machines to monitor and control tasks. Wireless-enabled embedded systems, real-time analytics, and machine learning inference sit at the core of IoT.

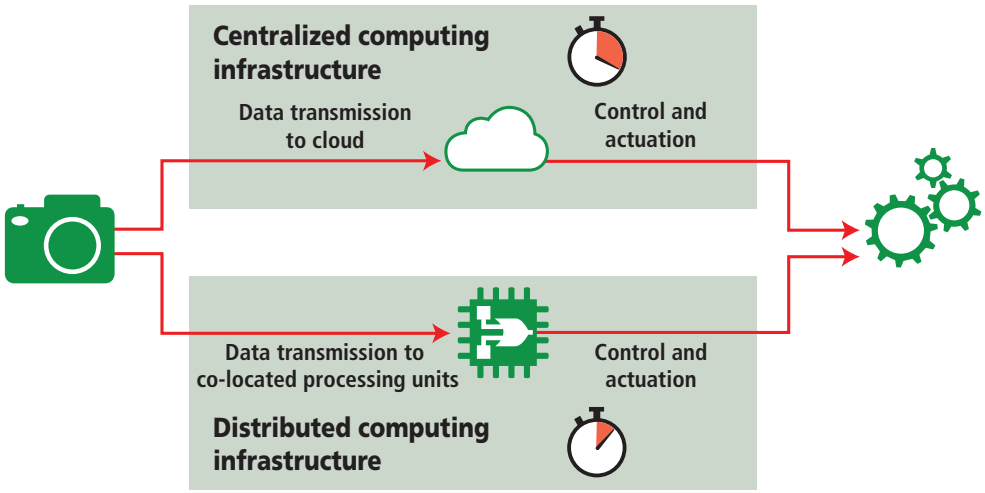

FIGURE 1: Edge computing optimizes response times and saves bandwidth.

A centralized storage and computation model does not offer an optimal approach for such applications, since it involves transmitting data to a central cloud server for processing and back to the devices for actuation. Edge computing addresses this by making compute

resources available close to the data sources (Figure 1).

Edge computing demands accelerated computational performance with accurate and predictable outcomes, along with strict power and footprint budgets. These limitations become especially complex when the applications involve embedded vision, since they require increased thermal headroom, higher resolution with multiple 4K/8K channels supported by batteries, and often integrate artificial intelligence (AI) and machine learning capabilities.

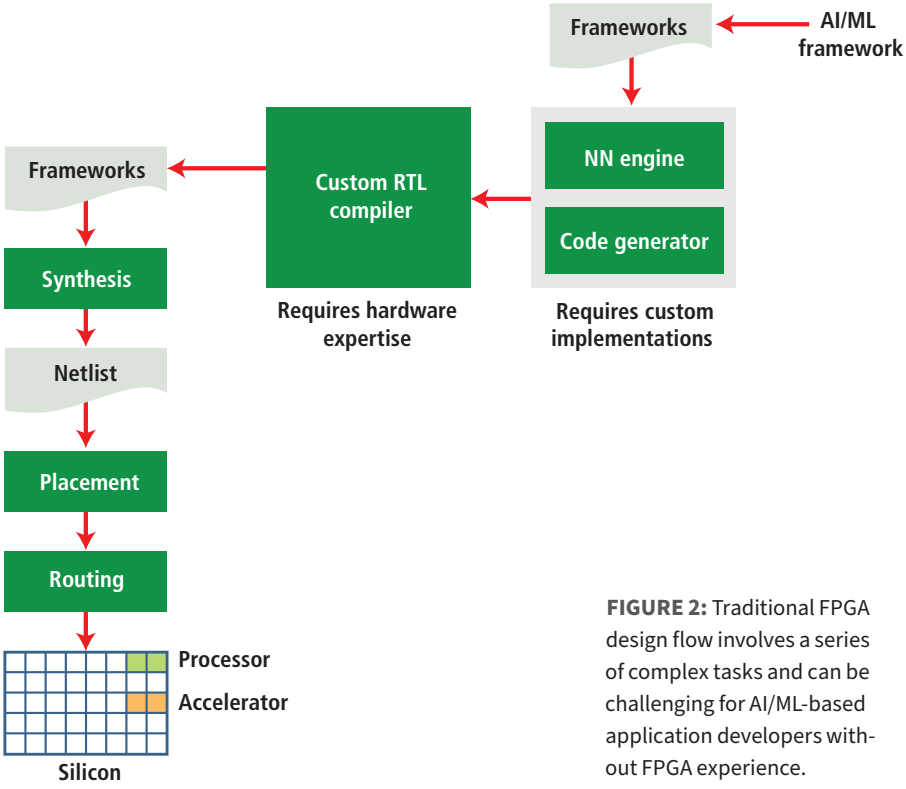

FIGURE 2: Traditional FPGA design flow involves a series of complex tasks and can be challenging for AI/ML-based application developers without FPGA experience.

Having an efficient edge computing system for smart vision systems requires flexible hardware that can support multiple interfaces for sensors such as MIPI and SLVS and transport, including HDMI, HDCP, USB, and SDI, with advanced image processing capabilities that allow for protocol conversions, filtering, and edge and depth detection at less than 5 W power consumption. Owing to their accelerated processing, reconfigurability, and energy efficiency at optimal cost, FPGAs present an affordable and powerful platform suitable for meeting edge computing requirements for embedded vision systems.

Compute Horsepower and Power Efficiency

Logic elements, an array of digital signal processing (DSP) interlocked with memory blocks, comprise the building blocks of an FPGA. Intrinsically parallel and well suited for specialized tasks that demand extensive parallelism during processing, hardware-programmable FPGAs accelerate the convolutional neural network (CNN) used for object detection and identification and other computer vision applications. FPGAs offer larger DSP capabilities than most CPUs and obviate the need for using external DSP elements, reducing total cost for a given functionality. Additionally, by interlinking memory with the core computing units in a distributed fashion, FPGAs bring processing closer to memory which contributes to optimized power utilization.

Edge devices may be deployed in remote locations with limited human access, and/or may require an uninterrupted power supply for critical applications such as aerial vision, medical vision, traffic monitoring, and other such automated processes. Even if a solution can offer high-performance in a small footprint, it may not necessarily be power-efficient.

FPGAs offer up to 30% lower power dissipation in vision-based machine learning applications as opposed to CPUs that work with a GPU. For a comparable throughput, FPGAs enhance thermal stability and optimize cooling costs. Devices based on non-volatile

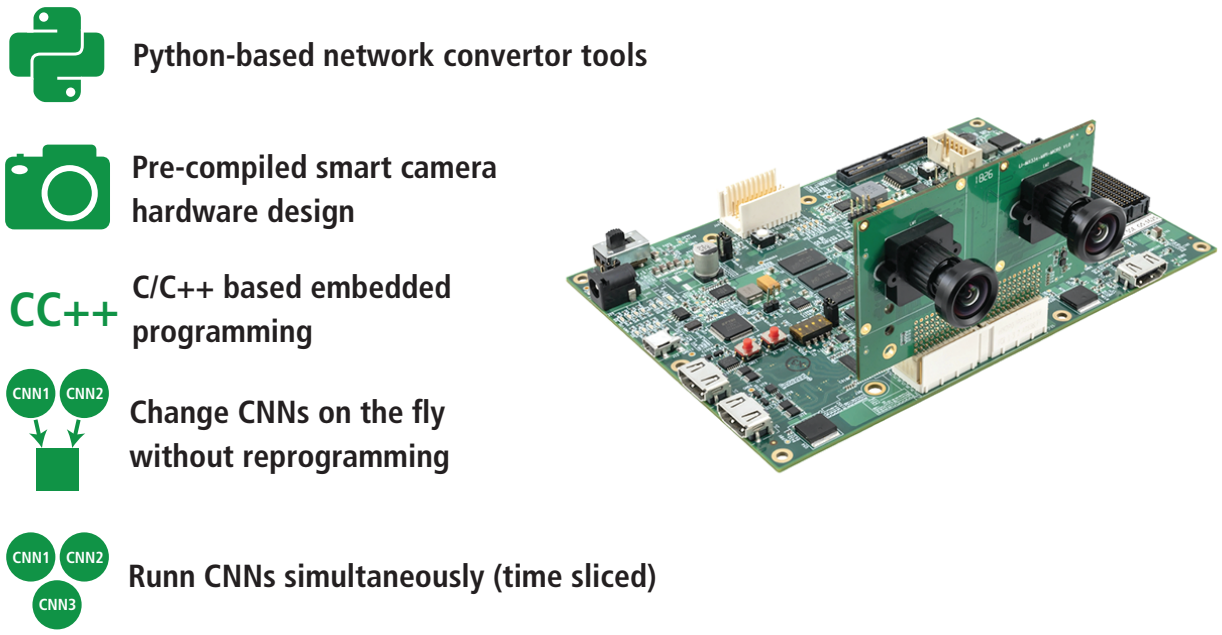

FIGURE 3. Microchip's VectorBlox accelerator SDK is part of its smart embedded vision systems initiative and includes video, imaging, and machine learning IP plus the tools for accelerating designs that require high performance in low-power, small form-factors

Flash/SONOS technology that feature transceivers architecturally optimized for mid-bandwidth applications inherently provide the necessary power efficiency.

Customization and Time to Market

Flexibility represents a more obvious benefit of deploying FPGAs in vision-based machine learning devices for three reasons. First, FPGAs integrate numerous diverse resources into one chipset such as hard and soft IP cores for camera sensor interfaces, control logic, compression algorithms, display and network interfaces, and an architecture favorable for neural networks. Second, by definition, FPGAs can be programmed either in part or whole while in the field. For example, if a developer were to update a deployed application system to latest versions of HDMI, MIPI CSI or even USB, it would be easier to do so on an FPGA than a custom built ASIC. It empowers one to build future-proof, scalable designs, conform to evolving standards, and reconfigure hardware for revised specifications.

If a MIPI-based image processing system necessitates an upgrade to support additional camera sensor interfaces, an FPGA can implement such a change with no modification to the system. This affords notable cost and time advantages. Furthermore, FPGAs are also scalable. If an algorithm or a function gets larger, FPGAs can be efficiently daisy-chained to adapt to the code.

Implementation Challenges and Requirements

While FPGAs represent a suitable option for the implementation of AI/machine learning-based functions at the edge, designing with an FPGA can be daunting and poses challenges especially for someone without prior FPGA experience.

A typical design flow involves working with one of the many available machine learning platforms to source tools, libraries, and resources to build a framework for a neural network and generate the associated code (Figure 2). The code compiles using a custom register transfer level (RTL) compiler followed by the traditional

FPGA design flow of RTL synthesis, netlist generation, and place and route. FPGA's programming model is inconsistent with the larger software development community. Although a wide range of platforms to develop a machine learning framework exist, each offers a unique design structure and requires custom implementation. There's also limited availability of application-specific evaluation platforms with an easy out-of-the-box experience in the FPGA industry. Investing in the necessary hardware to validate a CNN—particularly when one evaluates multiple platforms to determine the optimal option—also becomes imperative.

A software developer with no FPGA experience must be able to program a trained neural network on a hardware-free evaluation and validation platform, with access to multiple OS support. Meeting the programming challenges on FPGAs tailored around machine learning applications requires a unique combination of techniques and design flows. Such a combination must offer an extensive range of interoperable frameworks and abstracts hardware programming by allowing developers to code in C/C++ and use power-efficient neural networks.

One example of this approach can be seen in the software development kit (SDK) that Microchip (Chandler, AZ, USA) offers with its PolarFire FPGAs (Figure 3). Such kits enable the development of low-power, flexible overlay-based neural network applications without having to learn an FPGA design flow.

Kits that include a bit-accurate simulator enable users to validate the accuracy of the hardware while in the software environment. Kits should ideally also include neural network IP so that different network models can be loaded at run time and should also provide the flexibility to port multi-framework and multi-network solutions.

Security and reliability considerations must be made to ensure authentic, tamper-proof, and safe inference, particularly in applications like surveillance or drones. Updating the FPGA in real-time provides a fundamental advantage because susceptibilities can be addressed with new definitions. The FPGA must present competitive security features to ensure full design IP protection, secure data communications, and anti-tamper capabilities.

In an embedded vision scenario, deep learning inference schemes are typically part of a broader system that integrates camera sensor interfaces like MIPI and SLVS, an image signal processing unit, and transport interfaces like CoaXPress, HDMI, 10GigE Vision, SDI, and wireless connectivity.

An ideally-suited FPGA for such applications should be able to seamlessly support and integrate diverse protocols and interfaces with minimal developer effort as part of a total system solution.

Decentralized edge computing has gained popularity to overcome the shortcomings of long latency and bring compute resources closer to data sources. For embedded vision applications such as auto-piloting vehicles, surveillance, and medical imaging, cost-optimized mid-range FPGAs combine high-performance, design flexibility, and energy-efficiency.

While the complex programming models extend a significant challenge to the adoption of FPGAs, there has been an extensive effort from FPGA vendors to overcome traditional FPGA design flow challenges and bridge the gap to enable the creation of low-power FPGA-based embedded vision systems.

Apurva Peri is a senior engineer, product marketing, at Microchip (Chandler, AZ, USA)

Powering Embedded Vision with Image Sensors

Embedded vision defines systems that include a vision setup that controls and processes data without an external computer.

MARIE-CHARLOTTE LECLERC

New imaging applications are booming, from collaborative robots in Industry 4.0, to drones fighting fires or being used in agriculture, to biometric face recognition and point-of-care handheld medical devices at home. A key enabler in the emergence of these new applications is more accessibility for embedded vision. Embedded vision is not a new concept—it simply defines systems that include a vision setup that controls and processes data without an external computer. It has been widely used in industrial quality control in the form of what are generally referred to as "smart cameras."

A recent change is the availability of affordable hardware components developed for the consumer market, which has drastically reduced the bill-of-material and their size compared with computers. For example, small integrators/OEMs can find single-board computers or system-on-modules (SoMs), such as the NVIDIA Jetson, in low volume, whereas larger OEMS can directly supply image signal processors, such as Qualcomm's Snapdragon. At the software level, off-the-shelf libraries have made specific vision systems much faster to develop and easier to deploy, even in fairly low quantities.

The second change that is fueling the growth of embedded vision systems is the emergence of machine learning, which enables neural networks in the lab to be trained and then uploaded directly into the processor so that it can autonomously identify features and take decisions in real-time.

Providing solutions adapted to embedded vision is critically important to companies in the imaging industry that wish to target these high-growth applications. Image sensors, which directly affect the performance and design of embedded vision systems, play a major role for larger adoption and its key drivers can be summarized by the SWaP-C acronym: decreasing Size, Weight, Power and Cost. A strong accelerator for new uses of embedded vision is to meet market-acceptable price points, which comes with a strong constraint on the vision system cost.

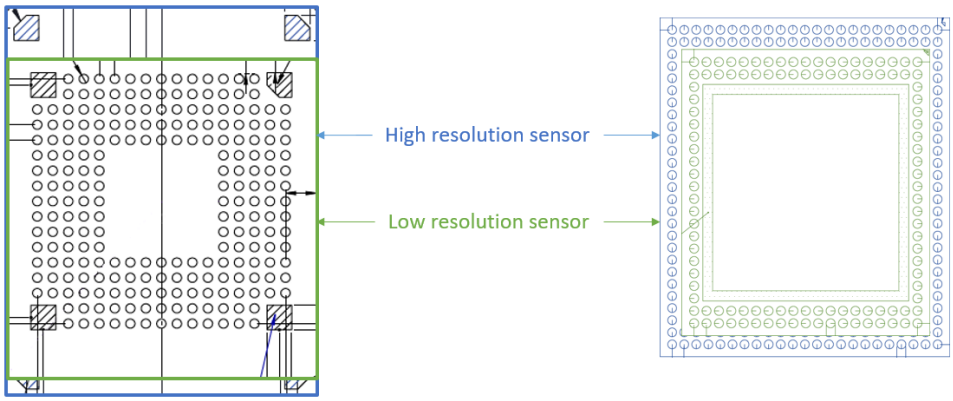

FIGURE 1. An image sensor platform can be designed to provide pin-to-pin compatibility (on the left) or footprint compatibility (on the right), enabling a unique PCB layout design. *(Images courtesy of Teledyne e2v.)*

FIGURE 2. New modules (on the right) allow direct connection to off-the-shelf processing boards (on the left) through flat cables without the need to design any additional boards.

Optics Cost Savings

The first way to cut vision system costs is to reduce footprint for two reasons:

- As the image sensor pixel size decreases, the intrinsic silicon cost shrinks because more chips can fit on the same wafer.
- The sensor can fit in smaller and lower-cost optics.

For image sensor manufacturers, this reduced optical cost has another impact on the design. As a general rule, the lower the optics cost, the less optimal the angle of incidence on the sensor is. Therefore, low-cost optics require the design of specific shifted microlenses positioned on top of the pixels so that they compensate for the distortions and focus light coming from wide angles.

Cost-Effective Interfaces

Aside from optical optimization, the interface also indirectly impacts vision system costs. The MIPI CSI-2 interface is the most suitable candidate to enable interface-induced cost savings as it was originally developed for the mobile industry by the MIPI Alliance. It has been broadly adopted by most ISPs, and the industrial markets have begun to adopt it as it offers a lean integration in the cost-effective system-on-chip (SoC) or SoM from NXP, NVIDIA, Qualcomm, Rockchip, Intel, and others. Designing a CMOS image sensor or imaging module with a MIPI CSI-2 interface provides a direct data transfer from the image sensor to the embedded system's host SoC or SoM without any intermediate converter bridge, saving cost and PCB surface, and that advantage is even stronger in embedded systems based on multiple sensors for 360° vision.

These benefits come with some constraints. The MIPI CSI-2 D-PHY standard, today widely used in the machine vision industry, relies on highly cost-effective flat cables with the drawback of a connection distance limited to 20 cm, which may not be optimal in remote head setups where the sensor is located farther from the host processor. This is often the case in traffic monitoring or surround-view applications. One solution for longer connection distance is placing additional repeater boards between the MIPI sensor board and the host processor—at the expense of miniaturization. Other solutions exist, coming not from the mobile industry but from the automotive industry: the FPD-Link III and MIPI CSI-2 A-PHY standards supporting coax or differential pair cables allow connection distances up to 15 m.

Reducing Development Costs

Rising development costs often present a challenge when investing in a new product. It can cost millions of dollars in non-recurring expenses (NREs) and create pressure on the time to market. For embedded vision, this pressure becomes even greater as modularity (which is defined by the ability to switch image sensors), is an important

value for integrators. Fortunately, the NREs can be limited by offering certain degrees of cross compatibility between sensors, for example; by defining families of components sharing the same pixel architecture to have steady electro-optical performances; by having common optical centers to share a single front mechanics; and a compatible PCB assembly, by means of footprint or pin-to-pin compatibility, to hasten evaluation, integration, and supply chain as illustrated in Figure 1.

Nowadays, developing embedded vision systems has become even faster and more affordable with the broad release of so-called modules and board-level solutions. These turnkey products usually consist of a ready-to-integrate sensor board that sometimes also includes a preprocessing chip, a mechanical front face, and/or a lens mount. These solutions benefit applications through their highly optimized footprint and standardized connectors, which enable direct connection to off-the-shelf processing boards such as NVIDIA Jetson or NXP i.MX units without the need to design or manufacture intermediate adapter boards. These modules or board-level solutions not only ease and hasten hardware developments by removing the need for PCB design and manufacturing, but also drastically shorten software developments, as they are provided along with their Video4Linux drivers most of the time. OEMs and vision system makers can therefore skip weeks of development in making the image sensor communicate with the host processor to instead focus on their differentiating software and overall system design. Optical modules can push that turnkey aspect one step further by also integrating the lenses inside the modules, providing a full package from the optics to the driver through to the sensor board and eliminating tasks related to lens assembly and testing.

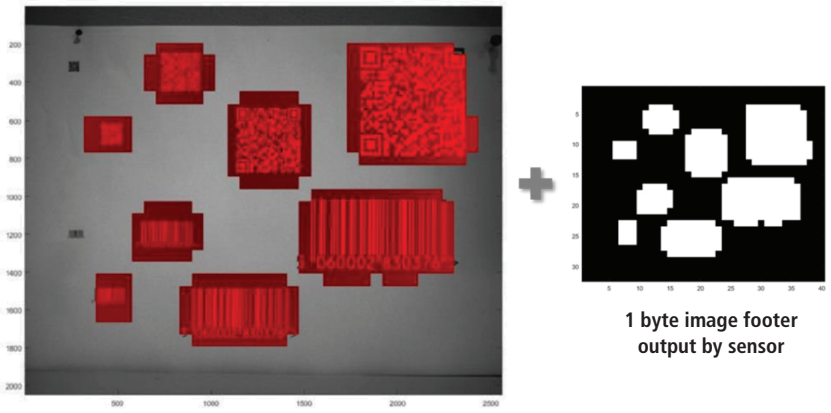

FIGURE 3. Automatic barcode location with the Teledyne e2v Snappy5M sensor.

Energy Efficiency for Enhanced Autonomy

Miniature battery-powered devices are the most obvious applications benefiting from embedded vision, as external computers prevent any portable applications. To decrease the systems' energy consumption, image sensors now include a multitude of features that allow system designers to save power.

From the sensor standpoint, there are multiple ways to decrease power consumption in an embedded vision system without decreasing the acquisition frame rate. The simplest way is at system level by minimizing the dynamic operation of the sensor by using (as much as possible) the sensor's standby and idle modes to reduce the sensor's power consumption. Standby mode switches off the sensor's analog circuit, reducing sensor's power consumption to a small percentage of the functional mode. The idle mode saves around half of the power consumption and can prepare the sensor to acquire images in microseconds.

Another way to save power is to design a sensor using more advanced lithography nodes. The smaller the technology node, the lower the voltage necessary to

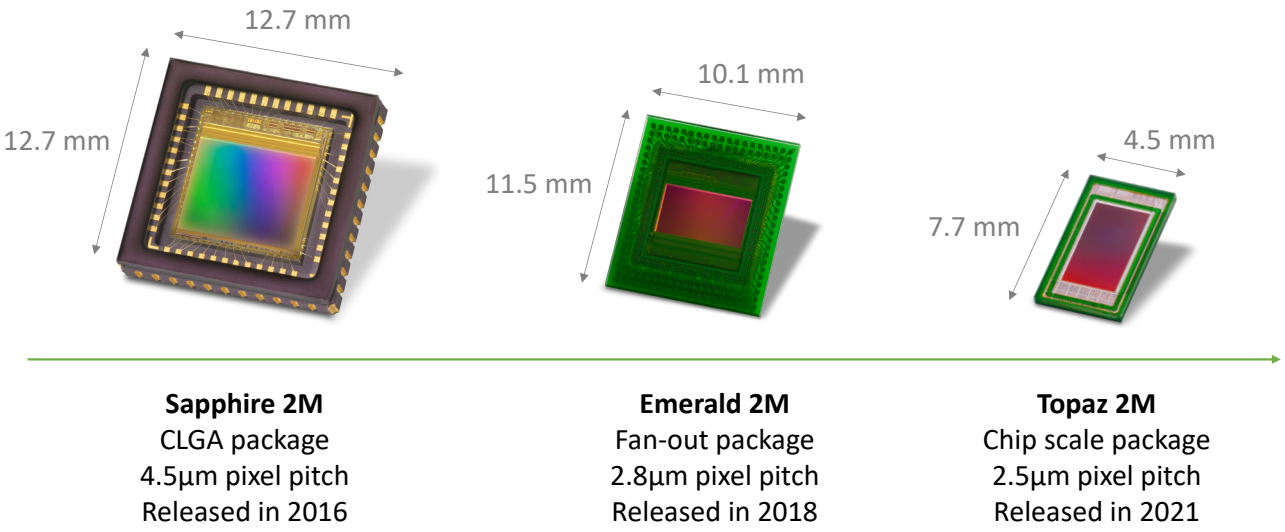

FIGURE 4: Typical evolution of image sensor footprint with package and pixel technology improvements since 2016.

switch the transistors and lowering the dynamic power consumption as it is proportional to voltage square: $P_{dynamic} \propto C \times V^2$. Therefore, pixels that were using 180-nm technology 10 years ago have shrunk by reducing the transistors to 110 nm and also decreased the voltages of the digital circuit from 1.8 to 1.2 V. In the next generation of sensors, the 65-nm technology node will be used to provide even more power savings for embedded vision applications.

Lastly, an image sensor can be appropriately chosen to reduce the energy consumption of the LEDs in certain conditions. Some embedded systems rely on active illumination, for example, to generate a 3D map, to freeze motion, or to simply increase the contrast by using sequentially pulsing specific wavelengths. In these cases, the image sensor can generate power savings by lowering the noise of the sensor when operating in light-starved situations. By lowering the sensor noise, engineers can decide either to reduce the current intensity or the number of LEDs integrated in the embedded vision system. In other conditions, where image capture and LED flash are triggered by an external event, choosing the appropriate sensor readout architecture can lead to significant power savings. Whereas conventional rolling shutter sensors require LEDs to be turned ON during the whole exposure of the frame, global shutter sensors allow it to turn ON the light for only a portion of frames. Switching from rolling to global shutter image sensors, therefore, induces lighting costs savings while still maintaining the noise as low as the CCD sensors used in microscopy if using in-pixel correlated double sampling.

On-Chip Functionalities Pave the Way for Application-Designed Vision Systems

An extreme extension of this embedded vision concept would lead us to a full customization of the image sensor integrating all the processing functions (SoC) in a 3D stacked fashion to optimize performance and power consumption. However, the cost of developing such a product would be tremendously high. While it is a custom sensor, it is not totally impossible to reach that level of integration in the long term. Today we are at an intermediary step, consisting of embedding specific functions directly into the sensor to reduce computational load and speed processing time.

For example, in barcode reading applications, Teledyne e2v has patented an embedded feature

directly on the sensor chip that contains a specific barcode identification algorithm to locate the position of the barcodes in each frame, so the ISP can focus on the these regions to process data more efficiently.

These functionalities are often specific and require a good understanding of a customer's application. As long as the application is sufficiently understood, other on-chip functionalities can be designed to optimize the embedded vision system.

Reducing the Weight and Footprint to Fit in the Smallest Spaces

A major requirement for embedded vision systems is to fit in small spaces or to be lightweight to fit within handheld devices and/or maximize battery-powered engines. That's why today most embedded vision systems use small optical format sensors with a limited resolution from 1 to 5 MPixels.

Reducing the dimensions of the pixel array is only the first way to reduce the footprint and weight of the image sensor. Today, the 65-nm process enables us to decrease the global shutter pixel pitch down to 2.5 µm without damaging electro-optical performance. Such manufacturing processes lead to products such as a full HD global shutter CMOS image sensor in the same format as in the mobile phone market, i.e. less than 1/3 in.

Another key technology for reducing sensor weight and footprint is to reduce dimensions of the package. Wafer-level packages have experienced fast growth in the market for a few years, especially for mobile, automotive, and medical applications. Compared with the classical Ceramic Land Grid Array (CLGA) packages used in the industrial market, the wafer-level fan-out packages and chip scale packages provide higher density connections, an excellent solution to the challenge of producing miniature and lightweight image sensors for embedded systems.

Looking to the future, we can expect another technology to bring further reductions in the size of sensors for embedded vision.

3D stacking is an innovative technique to make semiconductor components by manufacturing the different circuit blocks on separate wafers, and then stacking and interconnecting them with Cu-Cu connections and Through Silicon Vias (TSV). 3D stacking allows devices to be made with smaller footprints than conventional sensors because of layer overlap. In 3D stacked image sensors, the readout and processing blocks can be moved below the pixel array and row decoder. The footprint therefore decreases by the surface of the readout and processing blocks, while also bringing the possibility of adding extra processing power in the sensor to unload the image signal processor.

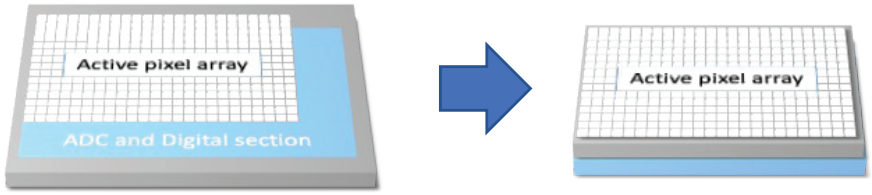

FIGURE 5: 3D chip stacking technology enables overlapping a pixel array, analog and digital circuit, and even adding extra layers of application-specific processing while reducing sensor footprint.

However, 3D stacking currently faces some challenges in order to be widely adopted on the image sensor market. First, this is an emerging technology; second, it's higher cost because of the additional process steps required that increase the silicon cost by more than three times compared with conventional technology wafers. Therefore, 3D stacking will be an option mostly

for high-performance or very small footprint embedded vision systems.

Embedded vision can be summarized as doing "lean" vision and can be applied by a number of companies including OEMs, system integrators, and standard camera manufacturers. "Embedded" is a generic term that is used in many applications, which makes it difficult to set a single list of specifications. However, several rules apply to optimize embedded vision systems, as the driving markets are generally not driven by state-of-the-art speed or sensitivity but rather by size, weight, power, and cost. The image sensor is a significant contributor to these parameters, and care is needed in the choice of the image sensor that will optimize overall embedded vision system performance. The right image sensors will offer more freedom for an embedded vision designer to reduce not only the bill of material but also the footprint of both illumination and optics. But even more than image sensors, the emergence of turnkey board-level solutions under the form of imaging modules paves the way toward further optimization of size, weight, power, and cost and a significant decrease of development cost and time, delivering affordable and deep learning-optimized image signal processors from the consumer market, without adding complexity.

Marie-Charlotte Leclerc *is a product manager, Teledyne e2v (Milpitas, CA, USA)*

How to Accelerate AI Deployment in Machine Vision Applications

Machine learning at the edge addresses applications too complex for rule-based vision but too simple to warrant investment in a full deep learning solution.

RETO WYSS

Traditional machine vision relies on analytical, rule-based algorithms to detect and parameterize defects that can be mathematically defined. In such applications, highly skilled systems developers and engineers evaluate each problem, apply a series of rules that can accomplish the task, and then program the system. To streamline the process, many vendors offer low-code and no-code solutions that help ease the process of tuning a set of analytical pattern matching, blob, edge, caliper, or other machine vision tools to meet application requirements. Despite these advances, rule-based solutions reach their limit when defects are difficult to define numerically or their appearance varies significantly.

As a result, ongoing development and maintenance of rule-based machine vision algorithms remains a challenge. It's often required due to part and process changes. Part changes can be caused by shrinking product lifecycles or component obsolescence, for example. Process changes may be required due to raw material or component variations from different suppliers, keeping up with technological advancements, or lighting changes in the production environment. This level of machine vision system maintenance relies on hard-to-find and expensive engineers with machine vision experience and skills.

Enter Deep Learning

A decade ago, deep learning was available only to specialized professionals with big budgets. However, advancements in theory, computer hardware such as GPUs, and data availability have recently led to its emergence in industrial machine vision applications. Deep learning excels in two areas: situations where

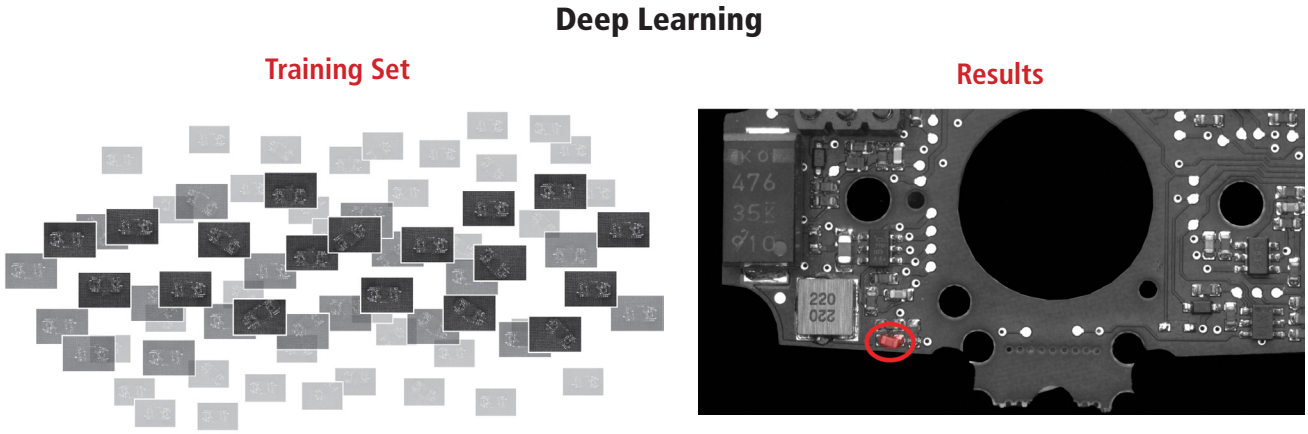

FIGURE 1. Deep learning is designed to automate complex and highly customized applications by processing large, detailed image sets, allowing users to quickly and efficiently distinguish between acceptable and unacceptable anomalies and deliver accurate results. *Photo Credits Cognex Corporation*

subjective decisions need to be made, such as those requiring human inspectors, and confusing scenes where identifying specific features in the image is difficult due to high complexity or extreme variability. Scenes with significant background noise—for example, leather products with texture—are a good fit for deep learning.

In contrast to rule-based machine vision, which relies on experts to develop new algorithms, deep learning relies on operators, line managers, and other subject matter experts to label images as good or bad and classify the types of defects present in an image. This approach eliminates the need for highly skilled machine vision specialists and reduces the size of the engineering crew required to deploy and maintain machine vision solutions. When something changes, anyone who knows what the defect looks like can retrain the model by recording and labeling new images.

Deep Learning Challenges

Deep learning toolkits enable people to deploy learning-based machine vision systems more easily, but obstacles remain. For example, most successful deep learning projects still require large budgets and specialized expertise from vision engineers and data scientists to initially set up the system. However, not all projects will deliver sufficient value to the operation that would justify a significant investment, limiting the ability of deep learning to meet requirements in such applications.

As with any machine vision application, image acquisition hardware plays a critical role in the success of a deep learning solution. A well-designed imaging system is required to perform image acquisition and collection. Reliable and repeatable imaging techniques must be able to clearly distinguish features or objects of interest.

Part presentation, illumination techniques, and image resolution play an important role in identifying

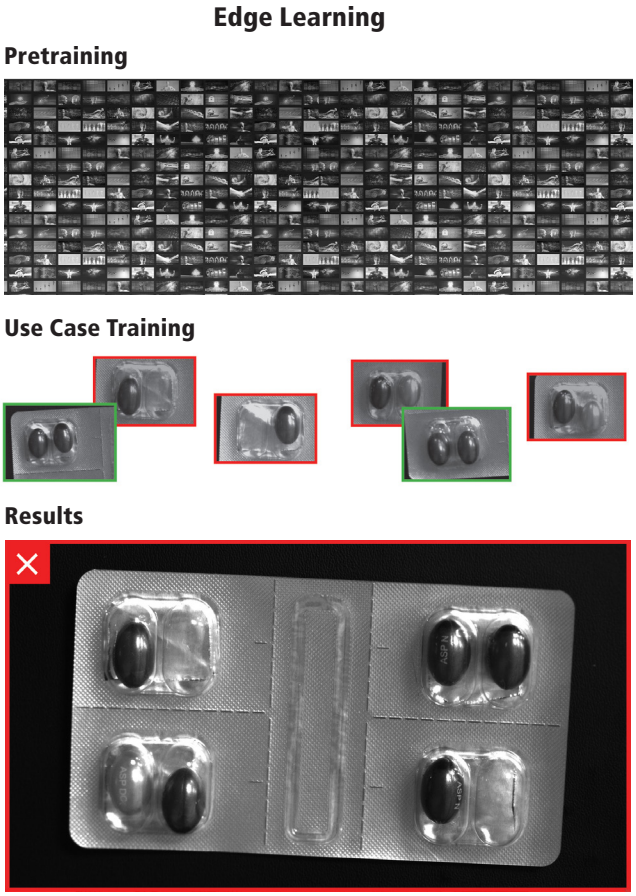

FIGURE 2. To ensure that edge learning networks can function efficiently on embedded machine vision systems, the images are resized or modified so that only the relevant regions of interest are examined.

the subtleties differentiating various classifications. And processing used for image analysis must be robust and powerful enough to handle typical production rates and algorithmic demands.

On the software side, model development can take a long time and require tagging of hundreds or thousands of images. Furthermore, obtaining images of defects can be challenging, particularly for prototype production lines that run small numbers of parts, as well as for consumer electronics and mobile devices that have very short production runs lasting a year or less. Such situations require frequent iteration. Moreover, highly automated production lines typically produce good parts with few defects. Consequently, it

may take several months of running the line to obtain a sample size large enough to generate a reliable model.

Edge Learning Minds the Gap

Considering all these challenges, many machine vision applications are too complex for a rule-based solution. Still, they don't warrant the time and resources required to develop a full-blown deep learning solution. To address this gap in machine vision application coverage between traditional rule-based and full deep learning solutions, hardware manufacturers have developed edge AI that runs on their embedded smart camera platforms.

Dubbed "edge learning," this type of deep learning utilizes a collection of preexisting algorithms that facilitate model training and subsequent image analysis directly on the device. Edge learning is a machine learning approach specifically tailored for industrial automation. It is trained in two steps: pretraining and specific use case training.

The first step is done by the edge learning supplier on a large dataset optimized for industrial automation. The pretrained tool is then embedded in a smart camera and shipped to the customer, who completes the second part of the training for their specific use case. This approach allows for faster training, requiring only a few images, and does not require a computer or GPU.

Image setup and acquisition also take less time because smart camera platforms combine multiple elements, such as sensor, optics, processor, and sometimes even illumination. This approach reduces hardware integration problems such as cabling to a PC and incorporating the inference engine, which can be time-consuming and increase the complexity of a machine learning system.

Edge Learning Benefits

Edge learning offers several benefits. It's much less costly to deploy than rule-based machine vision and deep learning solutions. It requires fewer images and takes less time to train and compute. It allows for faster production ramp-ups and product changeovers because training and production occur in the same place, on the same device.

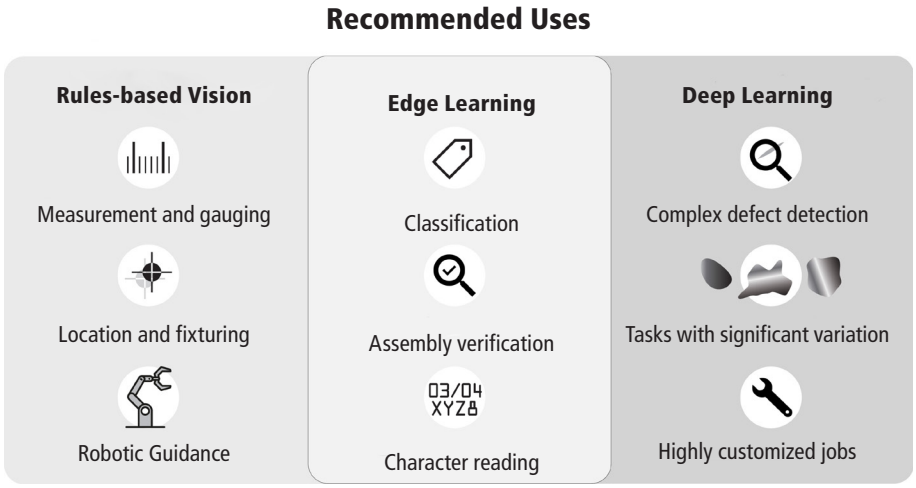

FIGURE 3. Edge learning fills the application gap between traditional rules-based machine vision and full deep learning solutions.

It should be noted, however, that the many benefits these edge learning-embedded smart cameras offer come at a cost. As a result, edge learning is not suitable for the most complex problems, but it can address a large portion of applications right out of the box.

Shorter Optimization Loop

Compared to deep learning, edge learning has a much shorter optimization loop and eliminates the need to send images to another device for labeling and retraining. Additionally, it optimizes workforce utilization

and reduces the long-term maintenance required for collecting and managing data.

Furthermore, edge learning is a viable option for automation as it doesn't require any prior knowledge of machine vision.Instead of relying on experts, edge learning allows operators and line engineers to label images for retraining of the system when part or process changes arise.

By enabling beginners and experts to quickly automate inspection tasks, edge learning benefits original equipment manufacturers (OEMs), machine builders, and end users alike.

Using edge learning, OEMs can more easily tackle challenging machine vision problems and empower their end-user customers. Edge learning enables end users to quickly address issues and add new products quickly, which minimizes the need to go back to the OEM and reduces the financial impact of after-sales support and service costs.

Meanwhile, system integrators can use edge learning to increase revenue by performing more feasibility studies in less time. Edge learning allows system integrators to reduce time spent on tasks such as image acquisition setup and machine vision tools selection so that they can quickly determine the feasibility of an application and win more business faster, while taking on more projects.

End-users can benefit from edge learning by automating many manual optical inspections or automation tasks that don't justify the investment of developing a sophisticated machine vision or deep learning system. Edge learning helps manufacturers more easily deal with part and process changes as they arise and iterate without developing new algorithms for each new generation of product.

Edge learning also can simplify existing rule-based machine vision applications and reduce costs associated with expensive image acquisition components, such as telecentric optics, illumination, or part handling systems. By simplifying or eliminating these costly components, a lower cost setup can often be achieved, with savings on image formation, fixturing, or complex image processing requirements.

Summary

Edge learning on embedded smart camera platforms offers a unique solution for many applications that are too challenging for conventional rule-based machine vision yet don't warrant the expense of investing in a full deep learning solution. Edge learning has proven to be more capable than traditional machine vision analytical tools in situations where human inspectors need to make difficult subjective decisions, for instance when identifying specific features in an image is difficult due to high complexity or extreme variability.

At the same time, edge learning is more cost-effective and user-friendly than traditional deep learning solutions, allowing more applications to be addressed economically. Edge learning tools can be trained using just a few images per class.

Ultimately, edge learning is another tool in the toolbox that can improve workforce utilization for OEMs, machine builders, system integrators, and end users.

Reto Wyss *is vice president of AI technology at Cognex Corporation. (Natick, MA, USA)*