

## Conquer the Challenge of Integrating Efficient Embedded Vision

[Electronic Design](#)

[Mike Thompson](#)

Mon, 2015-05-04 11:11



Giving electronic devices the ability to recognize objects and gestures, visually track things, distinguish you from me, and read our expressions and body language will change our world. Computer vision has been implemented on PCs and mainframes for decades, but only in recent years have we seen vision capabilities show up in embedded applications.

The embedding of vision into SoCs is being driven by the availability of high-performance processors, improvements in vision algorithms, and advances in process technology. However, including vision in a semiconductor device creates challenges. Foremost among them is being able to deliver the required performance while meeting the low-cost and power-consumption requirements needed for high-volume portable applications. This is difficult enough for designs that process VGA (640x480) resolution video, but market demand is moving to HD video—720p, 1080p, and beyond—forcing the need to process much more pixel data within the same power and cost budgets of previous-generation vision designs.

### **Embedded-Vision Applications**

Embedded vision is rapidly growing in emerging high-volume applications, such as automotive systems, wearables, gaming systems, surveillance, and augmented reality. The basic vision functionality embedded in these applications is object detection, gesture recognition, and mapping functions.

Related

[Q&A: Embedded Vision Alliance Founder Jeff Bier on Visual Intelligence's Future](#)

[What 3D Image-Capture Method Best Fits Your Needs?](#)

[Intelligent Imaging Yields Smarter Machine Vision Solutions](#)

Object detection lies at the heart of virtually all computer vision systems. Of all the visual tasks we might ask an SoC to perform, analyzing a scene and recognizing all of the constituent objects is the most challenging. Once detected, objects can be used as inputs for object-recognition tasks, such as instance or class recognition, which are methods for finding a specific face, car model, or unique pedestrian.

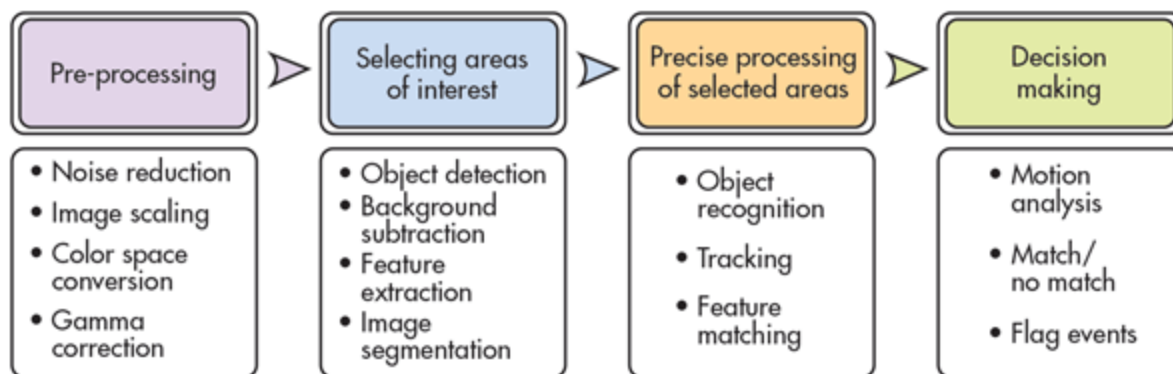
Gesture recognition is a class of applications that also involves object detection. Recognizing a gesture requires recognizing an object like a hand. However, it also requires tracking of the object in time to see if it changes in such a way that it would be interpreted as a gesture. Simultaneous Localization and Mapping (SLAM) takes the visual information that's been gathered and uses it to map an area to aid in navigation, or to make it easier to find things.

Over the next few years, adding detection and mapping capabilities to SoCs will change the world around us in significant ways. For example, our cars will become easier and safer to drive because of the richer awareness of the environment fostered by embedded vision. In addition, augmented reality will become more mainstream, making our entertainment experiences and games immersive by pulling us into the action. Innovative new products and software under development will enable the embedding of vision in electronic devices.

### Vision Performance Requirements

This is not to suggest that the challenges of embedded vision are simple to solve. An immense amount of computational capabilities are required to process a video frame to find an object. We take vision for granted because it's not a struggle to use our eyes to see and interact with our environment. While vision seems easy for most of us, the process is actually very complex and extremely challenging to implement in electronic devices.

Though applications of embedded-vision technologies vary, a typical vision system uses more or less the same sequence of distinct steps to process and analyze the image data. This sequence is referred to as a "vision pipeline" (see the figure).



The process looks simple, but just constructing a typical image pyramid (in the pre-processing step) for a VGA frame (640x480) requires 10-15 million instructions per frame. Multiplying that by 30 frames per second will require a processor capable of 300-450 MIPS to handle just this preliminary processing step, let alone the more advanced tasks required later in the pipeline.

This isn't a problem for a desktop processor with a GPU that costs hundreds of dollars and consumes watts of power, but becomes a totally different story for a resource-constrained embedded processor. The ultimate solution is a flexible, configurable, low-power platform with user-level programmability. However, a system that can't reliably and consistently track/count/catalog the objects it's programmed to detect is of little use. Highly accurate object detection and recognition is extremely important, so special care must be taken when designing this portion of the system.

### Convolutional Neural Networks

Although many algorithms are used for object detection, systems based on convolution neural networks (CNNs) are proving to be the most effective. CNN-based systems attempt to replicate how biological systems see, and they're designed to recognize visual patterns directly from pixel images with minimal pre-processing. Since 2012, algorithms that use CNNs on high-end systems have been winning computer-vision contests, such as the [ImageNet Large-Scale Visual Recognition Challenge](#), and recently, both Google and Microsoft achieved error rates of lower than 5%, which is better than humans. Although the concept of a neural network has been around for a couple of decades, recent advances significantly increased the performance of these "deep learning"

tems.

Currently, CNN-based architectures are mapped on CPU/GPU architectures, which aren't suitable for low-power and low-cost embedded products. CNN implementations on FPGAs exist, but they're not ideal for many embedded applications, partly due to the higher power consumption of this type of fabric. Also, a CNN's structure changes based on training results, so non-programmable hardcoded logic would greatly limit the usability, and thus isn't suitable for most vision applications.

To address the need for programmability, a DSP or a very long instruction word (VLIW) processor could be used to implement the CNN. Although some very powerful processor architectures exist in the embedded domain, these processors don't offer flexibility when the performance of a single core is insufficient, since multicore systems built up from these processors may struggle to meet performance requirements due to shared-memory bottlenecks. A better approach is to implement the CNN as a dedicated programmable engine that's optimized for efficient execution of convolutions and the associated data movement.

### **Embedded Vision Processors**

Using a dedicated programmable CNN engine and coupling it to a high-performance RISC CPU results in embedded-vision performance exceeding 1000 GOPS/W, as seen with the [Synopsys ARC EV processor family](#) and its ARC EV52 and EV54 processors. These processors are designed specifically for implementing embedded vision on an SoC that requires high accuracy and good quality of results. The processors have two or four general-purpose RISC cores coupled with up to eight specialized CNN processing elements. With this heterogeneous core combination, designers are able to build embedded-vision systems that achieve a balance of flexibility, high performance, and low power.

### **Summary**

Implementation of vision in SoCs is increasing rapidly, and is showing up in a broad range of applications. Advanced vision algorithms like CNNs and vision processors like Synopsys' EV family, which combine programmability with the efficiency of dedicated hardware, address the challenge of implementing embedded vision efficiently. By offering high object detection accuracy and good quality of results, along with power consumption of a few hundred milliwatts, these vision processors are changing the SoC paradigm. The changes that embedded vision will bring to our world will be profound, and begs the question: "How long will it be before your car no longer needs you at the steering wheel?"

**Source URL:** <http://electronicdesign.com/embedded/conquer-challenge-integrating-efficient-embedded-vision>