Technical Article

Maximizing Machine-learning Inference at the Edge

Pekka Varis

Employing machine learning and neural networks in factories improves applications like machine vision, automated guided vehicles (AGVs) and robotics by making them smarter, which improves operational efficiency. Embedded microcontrollers and processors have been used for decades in manufacturing to automate repetitive processes, but deploying machine-learning algorithms to embedded systems beyond research prototypes is still in its infancy.

Let’s take a look at one application of machine learning in a factory. Today, AGVs receive input from sensors such as cameras or radar. Software running on a processor analyzes the data and makes decisions that control the electric motors and move the AGV around the factory safely. In a constrained environment like a warehouse, classical machine-vision algorithms can be successful, but machine learning can outperform them because can adapt to slight changes in such environments and enable more human-like perception and classification, which is critical as factories evolve to meet three key needs:

- Reduced navigation space as warehouse density increases.
- The ability to classify objects on the factory floor, not just detect them.
- The coexistence of AGVs in a shared space with humans.

How machine learning enables Industry 4.0

Machine learning inference is a powerful tool for processing and understanding information, specifically in machine vision, where it has been shown to outperform not only classical algorithms, but also humans in terms of accuracy. Such capabilities are achieved through deep learning, which is a subset of machine learning that uses deep neural networks trained with a large amount of data.

To deploy machine- or deep-learning algorithms in an embedded system, the first step is to collect a data set of what requires classification or detection; for example, millions of images of obstacles that may exist on a factory floor, such as a person, a static machine, a robotic arm, shelving or boxes. You would then import this data set into an offline training program that learns to look for patterns or anomalies in the data. This step is not performed in real time, but is a computationally intense process that runs on servers or in the cloud. For machine vision, the output of this process is a trained network model that can be deployed on a processor at the edge of the network. For an AGV, machine-learning inference significantly improves the analysis of what its cameras see, which can then be used to take action, such as to avoid a box in its path.

Deploying machine learning in an industrial embedded system

The inference capabilities of a machine-learning-enabled processor apply the knowledge of a trained network to a given image or frame in a sequence of images. TI Sitara™ processors are embedded inference processors that enable equipment manufacturers to deploy machine-learning algorithms and automate applications like machine vision for AGVs.

Figure 1 illustrates a machine-learning example I ran on the TI AM5729 processor to show a semantic segmentation of pixels that identify three classes common in a factory: the AGV’s path (green), another vehicle (blue) and people (red).

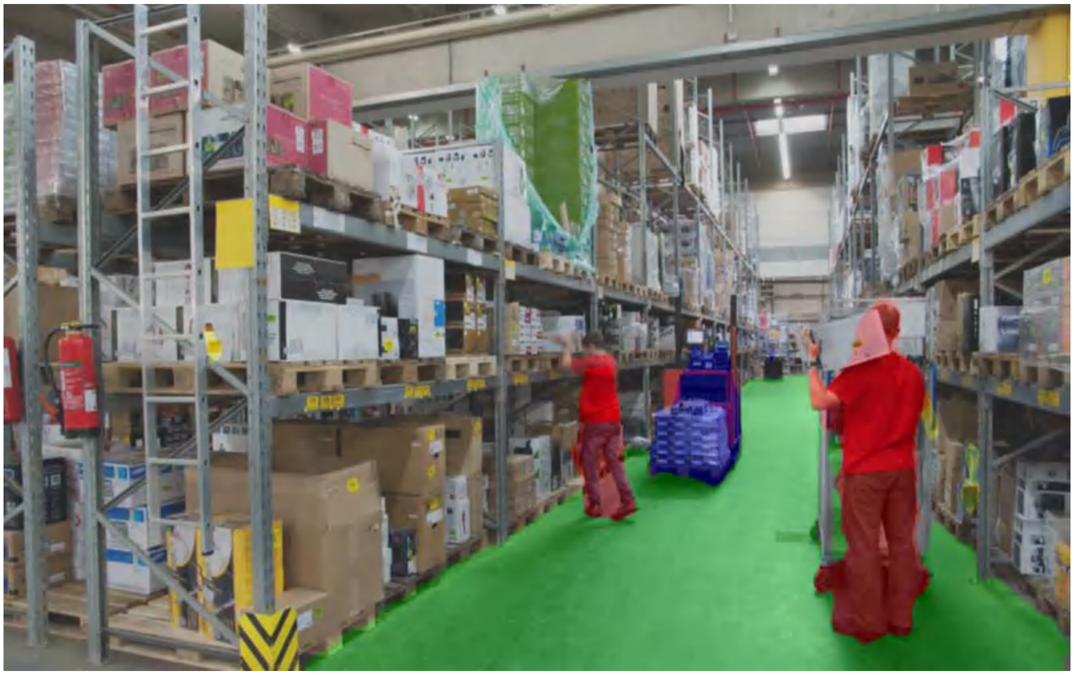

Figure 1. An example to illustrate the role of labeled data and machine learning

To illustrate how an AGV might operate, we used the jsegnet21v2 network from TI's processor software development kit (SDK) on an image applicable to AGVs in a warehouse setting. The example is trained on the cityscapes data set; for a production AGV, you would need to collect and label the pictures applicable to your environment to train the selected network.

Using the TI deep learning (TIDL) software framework, I deployed our algorithm on the Sitara AM5729 processor. The AM5729 processor ran the algorithm on a 1,024-by-512 pixel frame at about a dozen frames per second using the four embedded vision engine (EVE) cores on the device, and consumed less than 0.5 W of additional power. Each of the four EVE cores is a coprocessor able to perform 512-bit-wide vector multiply-accumulate operations that dominate the computational needs of neural networks. In real-time cyberphysical systems like AGVs, latency matters as well as throughput (frames per second). The frame processing latency on the AM5729 is approximately 250 ms (four frames in parallel), very likely sufficient to make decisions regarding the velocity of an AGV in a warehouse.

Reducing latency in machine-learning inference

In a typical example classification application using a network model like MobileNetv2, the high frame rate with low batch size achievable with the AM5729 translates to a 30%-40% decrease in inference latency per frame compared to the AM5749, which has two EVE cores. MobileNetv2 classification on 224-by-224-pixel images can run at 45 frames per second on the AM5729 processor. With less than 2 percentage point accuracy compromise, sparsification and TI's EVE-optimized deep-learning network model JacintoNet11, it is possible to improve the inference latency even further.

You can acquire or collect data and train your algorithms using these popular frameworks, and deploy your algorithms on Sitara AM5729 processors using the TIDL interface. Testing shows that processors with EVE cores underneath the inference run faster than on processors with just Arm® cores. Figure 2 compares the frames-per-second performance of two Arm® Cortex® A-15 processors against the AM5729 and AM5749 processors on some popular deep-learning networks. The Cortex A-15 performance is measured using Arm's NN 19.05 software, while the EVE performance uses the TIDL software framework in Processor SDK version 6.1.

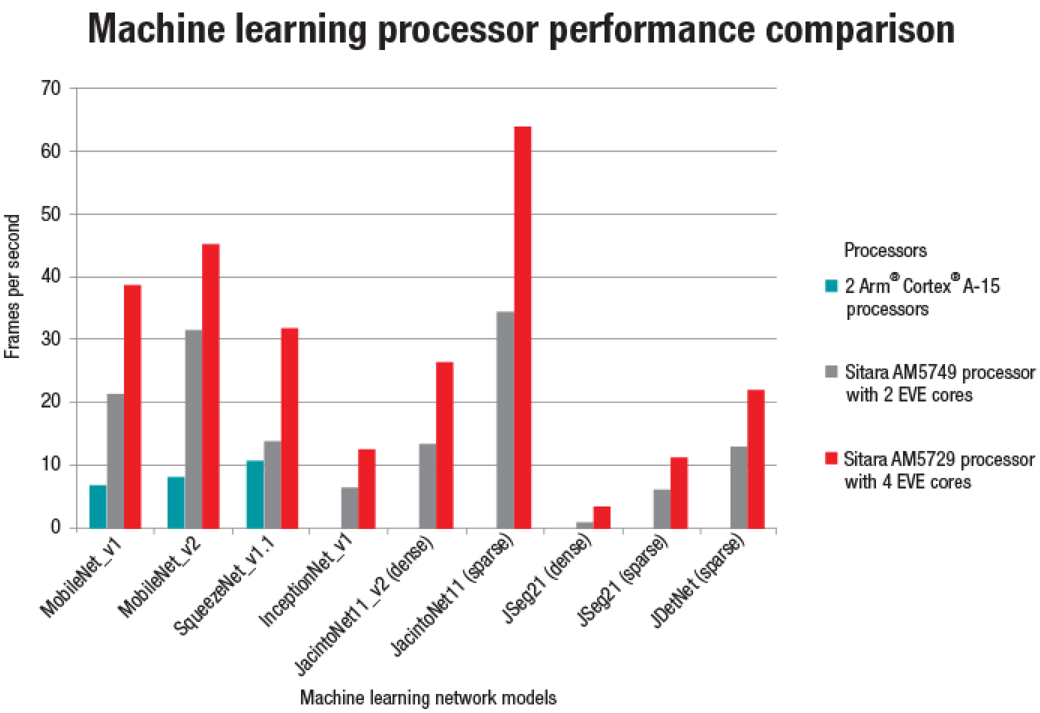

Figure 2. Frames-per-second performance of processors across various machine-learning networks

Today's distributed factories and breadth of applications will require machine-learning inference algorithms built on popular platforms like Sagemaker NEO and TensorFlow Lite that can run using the various network models listed in Figure 2. The ability to support these popular platforms is critical to manage the use of machine-learning inference everywhere as factory automation evolves.

Conclusion

Industrial embedded applications are about solving real-time use cases in the physical world, that can be proven with digital micro-benchmarks like frames-per-second throughput. In practice, the performance of the Sitara AM5729 processor enables the processing of typical camera inputs in real time, with frame rate, latency and power consumption relevant to multiple factory applications. To evaluate machine-learning functions with the AM5729 processor, the BeagleBone AI evaluation board offers a low-cost option for getting started, and the TMDSIDK572 IDK offers a full-featured, industrial-grade evaluation board for building and testing machine learning applications.

Additional resources

- Watch the video, "Machine learning hardware and software solutions at Texas Instruments."
- Get the Machine learning inference for embedded applications reference design.
- Download the TensorFlow Lite framework in the Processor SDK.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265