FPGAs Accelerate Deep Learning
Deep neural networks conquer image processing

Deep Learning is overtaking more and more tasks from classical algorithm-based image processing as the approach leads to better image processing results, even today, across industries in many applications. This two-part article illustrates state-of-the-art technologies for processing images and video as 2D and 3D data, as well as the resulting image processing applications for which deep neural networks such as CNNs (Convolutional Neural Networks) are especially well suited.

Content
Deep neural networks conquer image processing........1
Speed versus accuracy ..............................................1
FPGAs and SoCs for inference ........................................2
Optimize FPGA resources ...........................................3
More bandwidth by using preprocessing .........................4
Hard- and software closely interlinked ............................4
Result ............................................................................5

In image classification, deep learning is becoming more and more important, while image preprocessing, post-processing, and signal processing continue to be executed complementarily using conventional methods. If tasks can be accomplished exclusively, more simply, or with better results using deep learning, it then usurps classical image processing — especially in the case of aggravating variables such as reflective surfaces, poorly illuminated environments, varying illumination, or moving objects. Furthermore, classical algorithms are suited for cases involving exact determination of an object’s or defect’s position within an image, dimensional checking, code reading, or postprocessing.

Deep learning, in contrast, proves its value with very high reliability in identification rates, and will improve the quality of current image processing systems as well as open up new applications. Thus, the competition to be named the best solution centers on investment in algorithmic implementation versus training and preclassification time using clear marking of training images.

Deep learning differentiates between the neural network’s training and learning, implementation of the network — for example, on an FPGA — and inference, the execution of the network’s CNN algorithm on images with output of a classification result. In training, the network is modified via the weights (parameters) until the desired classification result is available. This is time consuming and assumes a high level of expertise. Deep learning requires large amounts of preclassified data and high processing power. The more data that is used for training, the higher the predictive accuracy for classification will be.

Speed versus accuracy

Defect and object classification using deep learning is currently a hotly debated industry topic. However, with machine vision’s special conditions surrounding rapid execution (inference) of CNNs with very low latencies, are the various processor technologies equally well suited for it?
High bandwidth, low heat output, and long-term availability are in demand, along with speed and real-time demands for which conventional CPUs or GPUs alone are hardly suitable. These represent more appropriate solutions for image processing tasks in the non-industrial sphere, where lower performance or accuracy sometimes suffice. If we merely compare technical aspects, these technology platforms show different performance values which exclude their use in applications with high demands. Thus, inference time of a GPU as a performance indicator averages lower than 8 MB/sec, at an assumed image size of 400x400 pixels, and a frame rate barely above 50 fps. However, GPUs represent the correct choice for neural network training, due to their very high data volumes. Since only the piece count 1 is typically needed for training, the hardware environment affects system costs very little. Here, commercial GPUs have emerged as the leading technology and are supported by almost every training software.

When selecting a suitable network, mid-sized networks often suffice for typical image processing applications when only a few characteristics are to be classified. These are sufficient for high speeds at slight reductions in accuracy. Competitive emphasis rests on ever larger yet slower networks that enable broad identification of various objects. At the same time, however, smaller, very efficient networks for classification tasks more typical of machine vision have come into existence. Here, it is clear to see that an acceptable loss of identification with a concurrent gain in data throughput is achievable by choosing appropriate network architectures for the application demands in question.

FPGAs and SoCs for inference

FPGAs as standalone processors or as SoCs together with ARM processors are best equipped for the demands upon inference of many image processing tasks, particularly of machine vision. FPGAs prove themselves with high parallelism of data calculation, guaranteed robust image acquisition and — in comparison to CPUs and GPUs — high processing power, image rate, and bandwidth. In so doing, CNNs on FPGAs classify at high throughput rates, something that fulfills particularly the time requirements of inline inspection.

The FPGA enables processing of image data — from acquisition to output — directly on a frame grabber or embedded Vision device without burdening the CPU, a quality which is particularly well suited for process-intensive applications like CNNs. Thus, smaller PCs without GPUs can be used, reducing overall system costs. The energy efficiency of FPGAs in the industrial temperature range is ten times higher in comparison to that of GPUs, making it ideal for embedded devices. This markedly expands the deep learning field of use with regard to Industry 4.0 as well as drones and autonomous driving.
For an FPGA, relocation into the fixed point sector will only lead to a negligible reduction in accuracy in comparison to that of GPUs, while the increased resources are available for larger network architectures or a higher data throughput. Thus it is possible to increase production speed in weld seam inspection or robotics, for example. Moreover, effective image preprocessing that reduces data enables use of smaller networks or FPGAs. These often suffice for simple classification tasks with few characteristics.

For high bandwidths, only FPGAs or GPUs can even be considered as usable technology. A comparison of the strengths and weaknesses of each shows that GPUs’ high processing accuracy and the resulting higher precision comes at the cost of clearly shorter availability and higher power consumption as well as lower data throughput. FPGA data processing is greater than that of comparable GPUs by a factor of 7.3. The increase in accuracy lies at 0.3% but means a 30% higher number of incorrectly classified images.

To compare system costs for an inference, processing systems based upon a programmable FPGA frame grabber, a specialized industry software solution with two GPUs, and two standard systems with an industrial and commercial GPU were used. Costs were determined using the components for a frame grabber (in this case, an industrial Camera Link solution), one or more GPUs, an industrial processor, and incidental software licenses. The juxtaposition did not take the output values for data throughput, bandwidth, and energy efficiency into account. The downside of supposedly more economically priced systems includes lower availability and poorer performance. Upgrading to industrial products (and GPUs as well) immediately leads to higher costs than FPGA-based systems. Industry solutions can result in high costs due to different licensing models (annual license, workstation license, or coupling) which in turn quickly exceed the hardware costs.

FPGAs’ processing power has developed to the extent that it has clearly surpassed the growth of conventional processors. Currently, the FPGA production process is being adjusted from 16nm to 7nm, resulting in a tenfold increase in logic reserves. In about five years, with adjustment to 3nm, an increase by a factor of 100 compared to today will be achieved which in turn will enable implementation of complex algorithms and automatic object classification within images. In order to achieve the high processing speed together with the high throughput rates and accuracy that many deep learning applications require, as well as for the implementation of large networks, higher-output FPGAs are necessary.

For processing speeds required in production, high-output frame grabbers and embedded Vision devices such as cameras and sensors with larger FPGAs are already available. Using more comprehensive FPGA resources, more complex architectures, and thus applications, can be processed. The higher data bandwidth enables processing of an entire image or additional image pre- and postprocessing on the FPGA. It is high enough to analyze the entire data output of a GigE Vision camera using deep learning, to name one example.

<table>
<thead>
<tr>
<th>System Cost</th>
<th>FPGA</th>
<th>Software</th>
<th>Industrial</th>
<th>Non-Industrial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame grabber</td>
<td>👎</td>
<td>🎩</td>
<td>🎩</td>
<td>🎩</td>
</tr>
<tr>
<td>GPU</td>
<td>👎</td>
<td>🎩</td>
<td>🎩</td>
<td>🎩</td>
</tr>
<tr>
<td>IPC</td>
<td>👎</td>
<td>🎩</td>
<td>🎩</td>
<td>🎩</td>
</tr>
<tr>
<td>Software license</td>
<td>👎</td>
<td>🎩</td>
<td>🎩</td>
<td>🎩</td>
</tr>
</tbody>
</table>

**Image 5: System cost comparison for a deep learning application**

**Image 6: Accuracy of processors with various calculation depths (floating point 32 bit versus fixed point/integer 16 and 8 bit)**

<table>
<thead>
<tr>
<th>Processor</th>
<th>FP-32</th>
<th>INT 16</th>
<th>INT 8</th>
<th>Diff. vs FP32</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG-16</td>
<td>86.6%</td>
<td>86.6%</td>
<td>86.4%</td>
<td>(0.2%)</td>
</tr>
<tr>
<td>GoogleNet</td>
<td>86.6%</td>
<td>86.5%</td>
<td>85.7%</td>
<td>(2.9%)</td>
</tr>
<tr>
<td>SqueezeNet</td>
<td>81.4%</td>
<td>81.4%</td>
<td>80.3%</td>
<td>(1.1%)</td>
</tr>
</tbody>
</table>

**Optimize FPGA resources**

Processing possibilities within the FPGA lie in direct proportion to available logic resources. These can be used not only for algorithmic description, but also for costly implementation methods, depth of calculation, or higher bandwidths via multiplication of the processing core. For deep learning, there are various methods for saving resources without reducing the quality of classification. One important method is image scaling that reduces the internal data throughput. Experience has shown that the depth of calculation only marginally affects the later identification accuracy. The reduction from 32 bit to 8 bit and from floating point to fixed point/integer enables the FPGA to invest the resources it has saved into larger network architectures.

A 32 bit floating point GPU’s higher computational accuracy is of little importance for deep learning inference, while 8 bit fixed point FPGAs achieve sufficiently precise identification accuracy for most deep learning applications with negligible error tolerance. As regards requirements for especially precise computational accuracy, 16 bit fixed point can be implemented on a larger FPGA as a resource compromise.
Identification of defects is a typical task in machine vision. If the error classes are known, training can be achieved using images with defects from production. With unknown images and the requirement to identify every deviation, however, the test pieces are compared with a “golden master”. In each of these cases, small networks are typically used that can detect error classes and their variants. Typical representatives of this are AlexNet, SqueezeNet, or MobileNet, which can be used as the basis for specific modifications and improvements to network architecture for an individual application. After training, FPGA implementation for inference follows. This represents a further opportunity to optimize resources and improve the quality of classification. To balance considerations of detection accuracy, implementation size and processing speed or bandwidth, small, fast networks are preferred in machine vision.

**More bandwidth by using preprocessing**

Frame grabbers or embedded Vision devices execute the entire image processing as central elements of the image processing system that include pre- and postprocessing, the actual CNN image classification, and synchronization of peripheral devices (lighting, sensors, and actuators) using signal control in real time. Using the VisualApplets graphical FPGA development environment, users are in a position to configure deep learning applications and image processing peripherals on their own on the FPGA using drag and drop, with no hardware programming experience whatsoever. Users see the representation and configuration of the neural network as a CNN operator within a data flow diagram. The specific CNN is connected between the camera operator as image source and the DMA operator as data transfer to the PC. The entire application design can be simulated in an ongoing manner as well as synthesized at the conclusion, and always runs in real-time operation at the defined speed with the lowest latencies possible.

If images are optimized using image processing prior to training, this procedure must likewise be executed during inference. The deep learning solution’s modular construction in VisualApplets enables the combination of preprocessing with image optimization, preselection of image details or even changes in resolution using the CNN processing core, and image postprocessing for ejecting bad parts. The CNN operator is configured in this case only with information regarding the network architecture and its weights.

Using the CNN operator, appropriate network architectures of varying sizes and complexities can be integrated and pretrained configuration parameters for network weights can be loaded for a variety of image processing applications. At the same time, information about networks and their parameters from third-party software such as the deep learning frameworks TensorFlow or Caffe can be imported. New weights are easy to load, as long as the network remains unchanged. Retrainings, such as those for new work pieces in production, occur with relatively little effort. Should the test environment or objects change, the retrained images can be reloaded via a new weights parameter set or as a new network (transfer learning).

**Hard- and software closely interlinked**

Implementation of deep learning applications on an FPGA is easy to manage and quick to implement with VisualApplets — and this as a hardware-programmed, real-time application. For inspecting reflective surfaces such as lacquer and metal using deep learning, an image processing solution is constructed within two to three weeks consisting of the microEnable 5 marathon deepVCL Camera Link frame grabber and VisualApplets. The sum total of all optimizations and the modular concept of the inference using preprocessing, deep learning execution, and postprocessing, enable improvements in classification without compromising performance. Users can simply continue to use their existing image processing system.
Result

For object and characteristic classification in image processing, more and more neural networks are catching on, especially for tasks with numerous characteristic or environmental variables, or those that are algorithmically difficult to describe. Deep learning’s strengths far outweigh its weaknesses. Relatively high investment in training is more than balanced out by reliability, speed, and possibility of results transfer. In the industrial field, FPGAs are the best choice due to their high processing power, low latencies, and power consumption, as well as long-term availability and prevalence even in embedded devices. Thus, FPGAs have become an interesting alternative from the point of view of their return on investment (ROI) when taking total costs of an image processing system into account.

Image 8: Processing of entire images, videos, and signals in real time using FPGAs