State of Art IoT and Edge Embedded Systems for Real-time Machine Vision Applications

Mahmoud Meribout, Senior Member, IEEE

Electrical Engineering & Computer Science Department, College of Engineering, Khalifa University of Science & Technology, UAE

Corresponding author: Mahmoud Meribout (e-mail: Mahmoud.meribout@ku.ac.ae).

This work was supported in part by the Khalifa University under Grant CIRA-86.

ABSTRACT IoT and edge devices dedicated to run machine vision algorithms are usually few years lagging currently available state-of-the-art technologies for hardware accelerators. This is mainly due to the non-negligible time delay required to implement and assess related algorithms. Among possible hardware platforms which are potentially being explored to handle real-time machine vision tasks, multi-core CPU and Graphical Processing Unit (GPU) platforms remain the most widely used ones over Field Programmable Gate Array (FPGA) and Application Specific Integrated Circuit (ASIC)-based platforms. This is mainly due to the availability of powerful and user friendly software development tools, in addition to their lower cost, and obviously their high computation power with reasonable form factor and power consumption. Nevertheless, the trend now is towards a System-On-Chip (SOC) processors which combine ASIC/FPGA accelerators with GPU/multicore CPUs. This paper presents different state of the art IoT and edge machine vision technologies along with their performance and limitations. It can be a good reference for researchers involved in designing state of the art IoT embedded systems for machine vision applications.

INDEX TERMS IoT, edge machine vision systems, Multicore CPU, GPU, FPGA, ASIC

I. INTRODUCTION

In recent years, processors are gaining increasingly computation power which has reached tens of Tera Floating Operating Operations per Second (TFLOPS) at relatively low power. This paved the way for the emergence of several IoT and edge machine vision systems which were not possible few years ago. This includes DNN cameras which can accurately perform complex tasks such as face recognition, environmental monitoring, and smart city promoting. For instance, presently very low payload vision-based navigation systems enable drones to navigate in GPS-denied environments such as indoor environments and within areas prone to high electromagnetic interference. Also, in autonomous car industry, current autopilot systems are very costly as they rely on expensive and bulky equipment such as Light Detection and Ranging (LIDAR) for navigation [1]. With the availability of powerful IoT/edge processors, having a machine vision system of an autonomous vehicle relying solely on few cameras with associated complex algorithms can be quite possible in the near future. According to [2], the global autonomous driving market is expected to reach 173.15 Billion USD by year 2030. Another very recent interesting application of IoT-edge based machine vision is to establish a defense line for COVID-19 in public places. Hence, simultaneous mask checking, counting the number of people, social distancing and temperature check is required to be automated using smart cameras. This can be an interesting substitute to current security, in addition it can accurately detect patients with COVID-19 symptoms [3]. These few instances of potential applications among others has led predicting that global smart camera market which was valued at USD 4.61 Billion in 2018 and is projected to reach USD 9.17 Billion by year 2026 [2]. It is also reported that by the end of 2021 around 1 Billion smart cameras will be deployed worldwide to handle wider spectrum of applications. This includes access control traffic monitoring, retail analytics, assets managements, and logistics [3][4][5][6].Thus heavy machine vision applications which used to be executed within edge devices such as DNN-based machine vision tasks are nowadays seamlessly moved towards IoT devices. This has the merit to drastically reduce the network traffic congestion and consequently to reduce security threats. A major distinguishable property of nowadays edge computing devices for machine vision applications is their ability to handle tasks received from several IoT devices. For instance, this can be done to allow access to a larger
reference database compared to what can be affordable within a single IoT device. This paved the way to an important emerging area being tackled by many researchers which is task offloading into edge devices and cloud servers. This is facilitated with the emerging 5G and 6G networks which yield higher throughput, lower latency, and lower network jitter for multimedia communication compared to former protocols. However, it is expected that security threats and network congestion with the increasing number of users will still remain pending challenges. Power consumption, which used to be one of the major issues few years ago, is effectively tackled with the emergence of new energy harvesting techniques. This includes low cost solar-based power supply in case of remote areas off-grid applications, and the continuous drop of power consumption of all kind of processors. This paper addresses the potential of different state-of-the-art processors such as multicore CPUs, GPU, ASIC and FPGA to handle machine vision tasks. It also suggests some future directions for tackling pending challenges. In the recent past, other review papers on IoT and edge processors targeting machine vision applications were reported [7][8][9]. One contribution of this paper is to address the most recent processors that use various technologies, along with their expected performance. For instance, the paper [7] mentioned NVIDIA’s GPU GTX-760 and Altera’s Stratix-4 FPGAs for various machine applications including stereovision, images registration and digital image correlation. In this paper, more recent processors, along with their performance in edge machine vision applications are described and analyzed. This includes the very recent Intel’s Stratix-10 and Aria-10 FPGAs chips and Xilinx’s Zynq Ultrascale for lower power and highly powerful deep learning-based machine vision applications. The other contribution of this paper, which to the authors’ best knowledge is not covered yet in the literature is to describe and analyze the main recent edge smart cameras which are currently available. Hence, unlike previous machine vision systems which were designed to be dedicated for a specific machine vision algorithm, nowadays edge hardware are flexible to host various machine vision applications using for instance DNN algorithms.

II. BACKGROUND

As per Wikipedia definition, edge computing consists to “push the frontier of computing applications, data, and services away from centralized node to the logical extremes of a network”. Thus, edge computing offers the advantage to yield lower latency and higher security protection than cloud computing (Figure 1). This is a very valuable feature for real-time machine vision tasks which are sought to be performed at the edge or IoT devices. This is facilitated by the continuous increase of the performance of processors to handle complex machine vision applications, such as real-time object detection and recognition. However, unlike cloud servers which have unlimited computation power and storage capacity, IoT and edge devices have an upper bound limit, due to the limited power consumption and limited footprint as they are mainly dedicated for portable applications. Compared to IoT machine vision devices, edge devices counterparts can process more than one input video stream. This requires them supporting very high video bus bandwidth such as Gigabit Ethernet to carry encoded video streams, such as H264, MPEG4, and even MPEG2 streams. This requires them to incorporate real-time video decoding modules to simultaneously handle several video channels. For instance the latest Advantech’s AIR-200 edge machine system, which is based on Intel’s Movidius Myriad-X Vision Processing Unit (VPU) can handle up to 120 fps video encode & decode of 1080 pixels, corresponding to 4 simultaneous video streams [10]. This also requires using real-time communication protocol (e.g. Real-time Protocol (RTP) over the Internet to reduce the network latency even further. In addition, edge machine vision platforms have the capability to process simultaneously several video streams (in the range of 10 approximately, versus thousands in cloud servers and only one in IoT devices).

![Fig. 1. Cloud computing versus edge computing.](image-url)

Figure 2 shows a typical hardware architecture of an embedded vision system, with it be an IoT or an edge device. In most of the cases, the input image needs to be completely stored in the frame memory, before the processing task starts, causing a latency delay of few ms. The actual image may be dispatched to PE’s local memories via direct access memory (DMA) channels, while another memory bank of the frame buffer is used to host the next image frame. This allows simultaneous capturing and processing of images. In some particular cases, such as image convolution, processing the input image on the fly using line memories (e.g. FIFO), the number of which is equal to the number of rows in the convolution mask is quiet possible using FPGA or even CPU/GPU processors if they have enough input ports to host the input images [11][12]. In terms of main processing unit, several processors use SIMD or systolic-like architectures where each PE has access to its local memory space to perform a same instruction as the other PEs [13][14]. While this computation model is efficient for low level convolution-type processing, it is not adequate for
other applications such as the ones requiring very high speed motion tracking within a relatively small region of the image, where a heavy workload is required for small number of PE, while several other PEs are ideal. In addition, this architecture limits the inter-PE communication to adjacent PEs, altering its performance for intermediate and high-level image processing tasks, such as image segmentation, features extraction, and geometric transforms, where frequent communication between non-adjacent PEs, is usually required. Memory based processing units such as Content Addressable Memory (CAM) were also used for both low and intermediate level image processing [15][16]. They offer the advantage to be highly dense, where each pixel image can be associated to a single PE. However, the PEs are limited to perform basic bit level logical operations hindering their performance for arithmetic intensive algorithms, such as the ones requiring intensive floating-point multiplications operations.

Thus, the need to significantly boost their performance has led today’s main processing platforms for embedded IoT-edge machine vision systems to be mainly multicore CPU, GPU, FPGA, ASIC-based, or a combination of some of them. For instance, some multi-core processors such as GAP9 processor integrates a multicore RISC-V processor, together with a DNN hardware accelerator for high speed machine vision application [17]. Similarly, several other ASIC or FPGA-based IoT or edge processors also include a single or multicore CPU for program flow control and also to handle tasks which are not supported by the ASIC or FPGA accelerators. Most CPU include multiply accumulate unit (MAC), zero overhead hardware loops and its manipulations which constitute a critical computation requirement in DNN and machine vision algorithms. As will be mentioned in Section 3, most systems comprises a communication module that may host new emerging low power communication protocols yielding relatively long communication range such as LPWAN (Low power WAN), in addition to legacy communication protocols such as PCIe, 100 Giga-Ethernet and WiFi since the IoT and edge communication devices can be either wired and/or wireless and to allow fast inter-chip communication. Video encoder/decoder are also usually embedded in most state of the art IoT and edge devices to handle average complexity encoding algorithms such as MPEG4 and H263, except MPEG-2 which is highly complex to be handled in IoT devices. Additional hardware modules which can be useful to end users, while they require a marginal additional hardware size may also be available. This may include single output sensors such as GPS (Global positioning system) and IMU (Inertial Measurement Unit) which can be useful in the perception systems of moving vehicles such as drones [18]. Unlike communication and video encoding/decoding modules which usually consume relatively low power and feature low die size, the main processing unit usually consumes the most power because of the high complex machine vision and DNN algorithms it must handle in real-time. This is the reason why a Power Management unit (PMU) is usually available to reduce the power consumption of the device using mainly three different techniques: Voltage scaling, frequency scaling or setting the processor in a specific mode of operation (e.g. sleep, deep sleep, and idle modes). Nevertheless, in a conventional system implementation, the standard upper ceiling of around 20–40 W, requires a heat sink and either a fan or substantial air flow for cooling, which is fine for edge devices, but not for IoT devices which are usually battery powered. When the core runs at about 7 W, the fan can be removed and there is less need for sophisticated management of hot spots. The video bus used to capture the input video stream is usually designed using high speed standard serial communication bus such as MIPI lane interface (for HD RGB cameras interfacing supporting up to 700 Million pixels/s), USB3.0 or Gigabit Ethernet to yield a data transfer of up to 5 and 100 Gbps respectively. Another important feature of these devices is the very high memory bandwidth, which can exceed 50 GB/s [19] to access to frame memory which is usually off-chip and DRAM-based. This is usually facilitated in most machine vision processors by embedding low power DRAM (LPDRAM) interfaces, such as LPDRAM-v4, for lowering power consumption during off-chip memory access. One more important feature of embedded IoT and edge machine vision systems when they are targeted for outdoor applications is their protection against harsh environments, such as dust and water. This is indicted by their ingress protection (IP) classification which can be up to the most recent IP69K standard for simultaneous high-pressure and high-temperature water (undersea application) [20]. However, this prevents the usage of an onboard cooling system which would consequently limit the computation power of the processing unit.

![Overall hardware architecture of an embedded machine vision system](image)
III. COMPUTATION MODELS IN MACHINE VISION SYSTEMS

Nowadays embedded IoT or edge machine vision systems are either modular where different phases of the algorithms are executed in different hardware modules or end-to-end based which are largely motivated by the development of artificial intelligence accelerators [21]-[25]. Modular designs allow different research teams with complementary background to work together. This includes the DAPRA project for building a perception system for autonomous cars [26], Junior project from Sanford [27], and Boss project from CMU [28]. Although the end-to-end approach has a potential to decrease the uncertainties facing modular designs, it still lacks mature, error free for real-life application [29].

A. Low, intermediate, and high-level image processing models

Image processing is a natural fit for data parallel processing which encouraged researchers to suggest different dedicated parallel hardware architectures. Based on their computation complexities, image processing algorithms can be categorized into three types: low, intermediate, and high-level processing. High level image processing mainly handles semantic data (e.g. classification) to estimate the semantic class. It has comparatively few data but very complex algorithms and would generally benefit from VLIW or RISC processor architectures. For instance, the results of vehicle tracking are explored for target range computation, range ate and forward collision warning [30][31]. These set of tasks which feature complex program flow make them a natural fit for a Multiple Instruction Multiple Data (MIMD) hardware architecture where the PEs execute different threads with marginal inter-PE communication. They can also be suitable to be implemented in powerful general-purpose CPU which explains why most IoT/edge machine vision processor, including GPU, usually comprise multicore CPUs. Pattern classification algorithm which is also a significant high level processing task can lend itself to an SIMD machine, which is the case for instance for the Support vector Machine (SVM). Similarly, Intermediate level processing such as image segmentation is also more tailored to MIMD architectures, where the parallelism is usually performed at thread level by allocating to each processor a region of the image and then proceed by aggregating the data to cover the whole image space. Nevertheless, few other image segmentation algorithms such as segment extraction can be suitable for an SIMD-like architecture where each processor is allocated to a given range in the parameter space [13][15]. In low level algorithms such as n x n convolutions, neighborhood pixels are processed using n buffers of n registers each (See Figure 3 in case of n = 3 to generate into the computation hardware the elements of the matrix a to i, where each element is multiplied by a constant value and the results are added altogether). In Figure 3(b) at each clock cycle, the FIFO memory outputs the pixel to both the first buffer A that holds the second row of the image and the first stage of a shift register. Both Buffer A and Buffer B delay the output by the image width. In this case, the frame is processed on the fly without the need to store it in an input frame grabber, unless there is the need to perform additional image processing tasks such as feature extraction or segmentation. Such hardware architecture model is mainly suitable for FPGA or ASIC-based processing units. The FIFOs should have the size of one column of the image and their number should be equal to the number of rows in the convolution mask, minus 1, which has the merit to yield 0 frames latency. SIMD architecture is another more frequently used alternative (e.g. GPU architecture) for low-level image processing where each processing element handles a different portion of the image while executing the same instruction as the other processing elements. While this hardware model which is featured in modern GPUs can efficiently increase the execution throughput, it still has several rooms of improvements. For instance, it does not easily explore the spatial or temporal pixel correlation which may exist between adjacent pixels and adjacent frames respectively. Stereovision is another very popular low-level image processing task, which explores the pixel correlation across multiple images which are simultaneously captured by several cameras using convolution operation on SIMD-like architectures. It is used in several applications, such as in autopilot driving system as a tangible alternative to LiDAR system despite yielding less accuracy (Lidar systems can provide a resolution of down-to 2 cm for a depth distance of 100 m), however it is much more cheaper since the actual cost of the LIDAR can easily exceed 50,000 US$.
In addition, stereovision camera can cover longer range of up to 2,000 m, compared to 200 m range of lidar [32]. Hardware architectures dedicated for Intermediary and high-level image processing tasks are complex and usually not scalable to handle a wide range of objects in the scene. For instance SIFT [4][5], Local Binary Pattern (LBP), Local Binary Pattern Histogram (LBPH), Haar transform, Gabor filter, Gabor Kernel fisher Analysis (KFA), and Principal Component Analysis (PCA) have proven to accurately detect and recognize complex objects such as faces and textures. Nevertheless, despite being successfully implemented on various parallel hardware architectures such as GPUs and FPGA these algorithms are still not mature enough for real-life applications since large data sets are usually required. Figure 4 summarizes the suitable hardware architecture that is suitable to each level of video processing. It also shows the required processing throughput and the suitable hardware architecture. Machine vision processors usually are designed to support all these architectures in a single chip.

**B. AI-accelerators for machine vision systems**

In order to overcome the non-scalability of hardware architectures dedicated for intermediary and high level image processing to handle a wide range of objects in the image space, AI-accelerators implementing either DNN or CNN inferring are now available for IoT or edge embedded machine vision devices. AI accelerators are capable to handle a descent number of objects, since they enable a rich vocabulary of objects (e.g. vehicles, types traffic signs, pedestrians, etc...) [33][34][35]. They have gained popularity from the fact they can be easily supported in a highly parallel SIMD or systolic architectures. While traditional ANN may comprise as low as 1 neuron and 1 hidden layer, DNN can comprise up to 1000s layers, which requires the usage of various optimization techniques to be fit in IoT or edge devices. Thus, DNN or CNN-based edge and IoT devices usually consist of the following optimizing sequence: (1) quantization which reduces the model’s precision from FP64 to lower resolutions (FP32, FP16, INT8) [36][37], (2) clustering which groups the weights of layers of the trained model into clusters, then the centroid of weight of each cluster are shared which reduces the complexity of a model by reducing the number of its unique weights, and (3) pruning or compression to remove the less relevant parameters in the model that have only a minor impact on the results corresponding to weights with very small value [38]. These steps have the advantages to reduce the complexity of the model without significantly affecting its accuracy [36][38]. For instance, Google’s Tensorflow is an end-to-end open source machine learning platform which contains more layers than other DNN models and require 64-bits floating point (FP64) precision [39] making it difficult to be handled in embedded IoT and edge machine vision devices. This has led researchers to design lower computation intensive inference DNN models which can be hosted into an edge platform. This includes TensorRT from NVIDIA [40] which performs tensor and layer fusion while reducing the memory requirements and to incorporate different optimizations algorithms tuned towards NVIDIA’s Jetson GPUs with the computation precision reduced to FP32, FP16, and even INT8, without introducing significant errors. Similarly, TFLite also reduces the memory requirement and number of bits per operands using quantization clustering and pruning [41] without significant loss of accuracy. In addition, it is platform independent making it adopted to any GPU architecture or even mobile phones (both IoS and android is smartphones). It can also be suitable for a fine grain pipeline architecture [42]. In [43], it was reported that the Tensor RT outperforms the TFLite by around 40% for real-time face recognition, but at the expense of high-power consumption when implemented on NVIDIA GPU. Thus, DNN features intensive low-level processing making it suitable for highly parallel architectures which are readily available nowadays, such as such as multicore-CPU or GPU using either SIMD or SIMT (single instruction Multiple threads) paradigms. This has led to the emergence of several DNN/CNN models which were designed for low power consumption, low computation power, and low memory capacity IoT and edge devices, while achieving good accuracy. This includes MobileNets family [44][45], SqueezeNets family [42], ShuffleNets family [45], and ESPNets family [41]. Nevertheless, SqueezNet
architecture remains the smallest and simplest type of DNN as it requires only 3 x 3 and 1 x 1 convolutions, pooling, and a fully connected layer, which is not required for feature extraction task [43]. Similarly to DNN, CNNs are too complex to run on IoT/edge devices with limited computing power. This has created large interest in designing efficient CNNs AlexNet [2] and GoogLeNet [14] CNN models have also become renowned for their exceptional performance in image recognition targeting IoT and edge machine vision tasks. In [41], it was shown that using the squared Euclidian distance as a pruning technique, 6.7 x and 1.5x compression were achieved for the MNIST and AlexNet networks respectively, while losing less than 2.2% accuracy of ImageNet accuracy. Another promising design approach for CNN targeting IoT and edge devices is to use the Neural Architecture Search (NAS) which aims to automate the design of neural networks by optimizing multiple different objectives together, like accuracy and efficiency, which is difficult for humans [3]. NAS methods have already achieved state-of-the-art results on image classification and object detection [34], outperforming the previous best manually designed networks.

IV. IoT and EDGE PROCESSORS FOR EMBEDDED MACHINE VISION SYSTEMS

In 1990s, FPGAs and DSPs were the technologies of choice for real-time image processing associated to low and middle complexity tasks [3]. This was due to the fact they host functional units such as multiply and accumulate (MAC), which was not the case of single core CPUs. For instance, TI’s OAP3530 DSP processor which includes a fixed point 520MHz-C64x core and an ARM’s 720 MHz Cortex-8 processor can run stereo vision algorithm using 7 x 7 pixel sliding window with a throughput of 8 fps [31], which however still remains weak for real-time applications. A higher caliber TI’s DSP processor, TMS320C6678, which features 8 x 1GHz cores CPUs which can be configured for both fixed point and floating operation was designed to handle high complex image processing algorithms, such as motion estimation, which however also still remains weak since the throughput for 128 x 128 image is only 9.79 fps [60]. Raspberry PI’s processor with its quad core Cortex A72 CPU @1.5 GHZ and 4GB RAM LPDDR4 in its last version (version 4) was also frequently used in machine vision applications targeting embedded systems such as drones and mobile robots. However, they are not the best choice for implementing massively parallel image processing algorithms such as DNN. Hence, since the emergence of GPU and multicore CPU processors in middle 2000s, DSP platforms have lost some ground for hosting real-time embedded machine vision applications. Whereas multicore CPUs spend most of their chip area on a couple of large, sophisticated cores, GPUs spend their chip area on a large number of clustered processing elements. Thus, GPU cores are much simpler than the cores in a CPU. For example, the GPU cores execute instructions in-order and have no bypassing network or branch predictor. Also, GPUs yields slightly lower accuracy since they do not support double precision FP arithmetic, which however is not required in most embedded machine vision applications. This has led to the emergence of several lower precision GPP processors , such as the Intel’s 14 nm Cascade Lake CPU, dedicate mainly for cloud computing using array of such chips, and which supports vector neural network instructions in 16- and 8-bits formats [61][62].

A. GPU-based IoT and edge machine vision systems

GPU processors were intensively suggested for IoT/edge machine vision platforms because of their powerful tensor cores that yields a tremendous speedup for the matrix computation which constitute one of the critical computations of machine vision and DNN/CNN inferencing. The basic design of a modern-day GPU contains clusters of cores in what is called by NVIDIA streaming multiprocessors (SMs). Each SM comprises several Streaming Processors (SP), which comprises an ALU that executes integer and floating-point arithmetic and logic operations. The number of cores in an SM is fixed to 32, however the number of SMs range from one for low-end GPUs to more than sixteen in high-end GPUs. L1 caches are available in each SM, however, they are much smaller than on a CPU. Additionally, all SMs in the GPU are connected to a shared L2 cache and the off-chip memory (GDDR) via a network on chip (NoC), as shown in Figure 5. To maximize the GPU overall performance, the shared L1 memory within an SM is organized into 32 banks (one bank per SM), so that all threads in a warp can access different memory banks in parallel. However, if two threads in a warp access different items in the same memory bank, a bank conflict occurs, and accesses to this bank are serialized, potentially hurting performance. GPUs achieve a very high degree of parallelism by having a single SM hosting several warps, holding up to 32 threads that can run in parallel at any point in time. Threads of a warp start executing at the same program address but have their private register state and program counters so that each thread is free to branch independently. However, when threads in the same warp follow a different execution path, threads are serialized by the hardware. Thus, GPU are based on single instruction multiple thread (SIMT) architectures and are designed to maximize the amount of parallel processing in the graphics pipeline. Unlike SIMD architecture where the processing engines typically execute the same instruction, SIMT architecture allows different threads to more readily follow divergent execution paths through a given thread program.
run on Jetson Xavier NX and Jetson AGX with INT8 segmentation [13]. In the Table the benchmarks were models which are suitable for image classification and edge GPUs to perform different kind of DNN inferring shows the performance of very more recent NVIDIA segmented based on their disparity. Table 1 below the image is divided into vertical regions that are yields a mid-level representation of the scene where performance of 42 frames/s could be achieved [16]. Another precedent version of these processors, Tx1 comprises 256-core supercomputer (Pascal GPU). powerful such as the NVIDIA’s TX2 processor which was recently tested in embedded processor, which comprises 1 to 4 core GPU to perform up to 500 Mpixels/s. the processor was initially designed for tablets and smartphones to accelerate graphics applications and then extended to host embedded IoT and edge machine vision tasks, such as real-time face recognition using an additional quadcore i7 processor [19].

**B. Multicore CPUs-based IoT and edge machine vision systems**

The limitations of multicore Superscalar processors are prominent as the difficulty of scheduling instruction becomes complex. The intrinsic parallelism in the instruction stream of machine vision algorithms, complexity, and the branch instruction issue get resolved by a higher instruction set architecture namely the Very Long Instruction Word (VLIW) or VLIW Machines. In this architecture several independent operations are grouped together in a single VLIW instruction and are initialized in the same clock cycle. This has the advantage to reduce the hardware complexity and power consumption, as well as the simple instruction dependency check. Thus Multicore CPUs targeting IoT and edge machine vision systems are usually optimized for single thread performance and are usually equipped with SIMD vector instructions, such as MMX, SSE and AVX vector extensions that usually equipped for single thread performance and are usually optimized for single thread performance and are usually equipped with SIMD vector instructions, such as MMX, SSE and AVX vector extensions that usually operate on 64 to 1024 bits data elements, to further enhance their performance when handling parallel algorithms. Several low power multicore CPUs were designed for real-time IoT and edge device. For instance, very recently, Geenwaves-technologies Company offers a low power and low latency multicore RISC-V CPU (8 CPU) DNN hardware accelerators, namely GAP9 processor which features 150.8 GOPS and a very low power of 0.33 mW/GO. It can operate at a maximal frequency of 400 MHz to manipulate 8 to 64 bits fixed and floating point precision, while on Jetson Nano and Jetson TX2 they were run on FP16 precision. Also, Jeston Xavier NX and Jetson TX2 worked at their high-power rating of 15 W, while Jetson Nano and Jetson AGX consumed 10 W and 30 W respectively. Nevertheless, the figures in the table demonstrate the potential of these processors to be used in IoT/edge embedded machine vision devices. Another GPU processor which was used for edge machine vision tasks is NVIDIA’s Ampere 100 Tensor core GPU [28]. It can be partitioned into up to 7 GPU instances to dynamically adjust to shifting demands, fully isolated with their own high-bandwidth memory cache, and compute cores. It can perform up to 9.1 TFLOPS FP64 or 19.5 TFLOPS FP32 or up to 1248 TOPS INT9 Tensor core. It comprises up to 80 GB memory. However, the processor consumes mor than 100 W. ARM’s 500 MHz Mali-400 MP2 is another available GPU processor, which comprises 1 to 4 core GPU to perform up to 500 Mpiixels/s. the processor was initially designed for tablets and smartphones to accelerate graphics applications and then extended to host embedded IoT and edge machine vision tasks, such as real-time face recognition using an additional quadcore i7 processor [19].

**Fig. 5.** A GPU consisting of 14 SMs and 5 memory partitions (gddr) with L2 cache, all connected by a network on chip. This specific configuration can be found in an NVIDIA GTX 470

Hence SIMD processing represents a functional subset of SIMT processing regime. A CPU (single or multicore) is frequently provided for GPUs to ensure their control and also to execute algorithms which are not adequate for SIMT/SIMD architectures. The CPU-GPU integration into a single chip is provided in most GPU processors to overcome the limited bandwidth of PCIe, which was used to transfer the data between the CPU and GPU in older GPU processors. In addition, both the CPU and GPU share the same main memory, allowing both CPU and GPU to have more opportunities for fine-grained cooperation [23]. Among the GPU processors recently used in IoT/edge cameras, one can mention the recent NVIDIA Jetson AGX Xavier processor which comprises 512-core Volta GPU It also includes 64 Tensor cores, an 8 core ARM v8.2 64-Bit CPU and 32 GB 256-Bit LPDDR4x with a bandwidth of up to 137 GB/s to yield up to 32 TOPS of AI performance. The processor can be configured to consume 10 W, 15W, or 30 W based on the target performance. This is the latest NVIDIA GPU processor which followed other ones almost 20 x less powerful such as the NVIDIA’s TX2 processor which comprises 256-core supercomputer (Pascal GPU). Another precedent version of these processors, Tx1 processor, was recently tested in embedded autonomous car to implement the stereo matching algorithm to perform 3D mapping algorithm where a performance of 42 frames/s could be achieved [16] yields a mid-level representation of the scene where the image is divided into vertical regions that are segmented based on their disparity. Table 1 below shows the performance of very more recent NVIDIA edge GPUs to perform different kind of DNN inferring models which are suitable for image classification and object detection, pose estimation, and semantic segmentation [13]. In the Table the benchmarks were run on Jetson Xavier NX and Jetson AGX with INT8...
The processor offers an 8-bit CPI (Camera Parallel Interface) and combines state of the art 22nm FD-SOI semiconductor process. It also includes security modules such as AES128/256 cryptography and a Physically Unclonable Function (PUF) that allows the device to be uniquely and securely identified. The processor is dedicated to process small size images (160 x 160 image) where MobileNet V1 network can run in just 12 ms for a power consumption of 806 mW/frame/s [7]. The processor is dedicated for battery powered IoT cameras such as people counting, and attention awareness to run for years with a single AA battery using adjustable frequency and voltage domains and automatic clock gating. The built-in hardware convolution (HWCE) module which is a co-processor dedicated for accelerated computation of CNN algorithms allows to run in one clock cycle a single 5 x 5 or 4 x 7 convolution using 16 bit weights and 16 bit pixels or up to four simultaneous 5 x 5 or 4 x 7 convolutions using 4 bit weight and 16 bit pixels. Using this processor, it was shown that the processor can perform face recognition, with 1 second latency, using only 1.5 Mb of weights with 16-bit fixed point integer computation. Intel Company is another active player in providing machine vision IoT and edge processors. For instance, the 700 MHz- Myriad X VPU, which features a neural compute engine, a dedicated hardware accelerator for deep neural network inference, is designed to handle edge machine vision applications such as autonomous navigation, Virtual/Augmented reality (VR and AR), and smart camera [15]. The 16 nm technology processor which consumes as low as 2.5 W features 16 programmable 128-bit VLIW vector processors, a neural compute engine (DNN Accelerator) to perform 1 TOPS of DNN inferencing. It also comprises an enhanced vision accelerator, a 4 K real-time video encoder (H264/H265), 8 MIPI lanes to connect to 8 high speed cameras, and supports LPDDR4 interface which are attractive attributes for an edge computing device. The vision accelerator is modular and comprises 20 dedicated hardware modules to perform tasks such as stereo depth and motion estimation in real-time (@ 60 Hz frame rate) by processing the image pixels of up to 6 cameras input (720p) on the fly, without storage. These features have led the processor to achieve an aggregated performance of up to 4 TOPS for 2.5 W power consumption. Very recently, a PCIe board, namely AD-VEGA-340, which includes 8 Movidius Myriad X VPU processors was suggested to handle complex AI algorithms at the edge. Another similar board that comprises also 8 of such processors, interconnected with each through via 4 PCIe/USB3.0 interfaces to consume less than 30 W was suggested in [12] (Figure 6). The board consumes approximately 25 W and comprise a dynamic interconnection engine that interconnects the processors according to various topologies.

Very recently, a high performance Industrial machine vision edge system that 16 of such processor, is commercially available by AAEON company to support up to 200 IP cameras to handle various machine vision applications at the , including automated warehouse and logistical processing, production line monitoring, and door security [33]. Most CPUs in the aforementioned processors supports VLIW instruction set for the reasons invoked earlier. However, this requires complex compilers, and an increased program size, which may be prone to unscheduled events such as cash miss which could lead to a stall of the entire processor. Texas Instruments is another potential IoT-edge machine vision processor player targeting mainly perception systems for autonomous cars, through its very recent 2GHz Jacinto TDA4x machine vision edge processor that yields an impressive 10 TOPS/W. The processor can be interfaced to up 8 MP FPGA link capture streams with real-time processing and analytics. It comprises an array of pre and postprocessing engines for neural network and is able to accurately identify lanes lines of various types and colors with the recognition curvature radius of at least 5 m. It also allows multisensory fusion using Lidar, IMU, and ultrasonic sensor information to allow high precision trajectory calculation [14]. The processor comprises a single DSP C7x processor to perform 80 GFLOPs or 256 GOPS, with neural network accelerator via matrix multiply accelerators (MMA) to perform up to 8 TOPs at 1 GHz, a duo DSP core C66x DSP, and Integrated imaging signal pipe (VPAC) with image signal processor (ISP), a hardware engines for optical flow & stereo disparity (DMPAC), and the latest dual core ARM cortex A72. It also comprises imagination 8XE series GPU. The multiple algorithms namely motion segmentation, depth estimation; multiple object detection (pedestrian, cyclist, vehicle etc.), semantic segmentation, parking spot detection and visual localization are performed on DSP cores with MMA (matrix multiply acceleration).

C. ASIC- based embedded IoT/edge machine vision systems
In spite of their high performance for both IoT and edge-based machine applications, GPUs and multicore CPUs processors are still relatively large and consume relatively high power since they are the result of scaling down an architecture which was originally dedicated for desktop computers to perform 3D graphics. In addition, while GPU and multicore CPUs excel for dense floating point computation, researchers have reported higher throughput and lower power consumption, at the expense of a reasonable drop of accuracy, with custom hardware using either FPGA or ASIC using low precision fixed-point quantization [15]. In addition, ASICs and FPGAs technologies are preferable in case of sparse data. For instance, in [27], an array of bit-level functional units which dynamically fuse to match the precision of individual DNN layers was only 16% slower than NVIDIA’s Titan Xp for implementing AlexNet, while consuming only 900 mW vs 250 W for the titan Xp which uses 8 bits instruction. These factors have led some companies to build their own ASIC and FPGA-based machine vision chips. This includes the recent ASIC SOC from Google’s Tensor Processing Unit (TPU) [37] and Intel’s very recent 10 nm Neural network Processor (NPN)-I-1000 [28] which can yield, an impressive performance of 4.8 TOPS/W. The processor comprises an array of inference compute engine (ICE) and two Sony Cove (SNC) IA cores. Each ICE engine consists itself of and array of Deep Learning (DL) compute grid engine to perform FP16, INT8, and INT4/2/1, a high througput 5 VLIW 512 bits programmable vector processor featuring extended neural network support, and a built in 4 Mb SRAM. Even though, the chip is dedicated for cloud computing (e.g. Facebook cloud using thousands of such chips), it can be scaled down to support IoT and edge machine vision algorithms since it consumes very less power per TOPs. Google’s latest 787 MHz TPU version#3 uses four 128 x 128 systolic arrays of MAC (namely Gigantic systolic multiply-accumulate, MXU) to yield an impressive 15-30x and 30x higher performance-per-watt over traditional CPU and GPU processors respectively [8]. This systolic array, which needs to be interfaced to a host CPU through two PCIe 3.0 x8 edge connector for an aggregate 16 GB/s of bidirectional bandwidth, is reused across the layers of the inference network, thereby boosting performance and energy efficiency of the accelerator. It consumes around 40 W which is readily available in the PCIe bus and delivers around 92 TOPS for 8 bits integer operations. Currently, this ASIC chip is not commercially available for developers as it is mainly used for remote access to Google cloud. Even though the chip is mainly dedicated for training in the cloud because of its large power consumption, requiring cumbersome liquid cooling [11], it can also be downscaled to so be implemented into IoT and edge machine vision systems because of its low power density. These attributes paved the way to a recent Google’s Edge TPU IC which performs inferring for machine vision tasks at the edge to yield 4 TOPS using 0.5 W for each TOPS [12] (Figure 7). The accelerator contains a 2D array of processing elements (PE), where Each PE embodies a single or multiple cores, each core having multiple compute lanes with multiple MAC units operating in single-instruction multiple-data (SIMD) style; however the PEs can work in systolic manner. Each PE has a PE memory shared across the cores, and each core is equipped by a dedicated core memory. The device is available as an USB accelerator that can be plugged into a desktop computer or added to a Linux-based Dev Board [13]. It is is one of the latest boards comprising edge TPU  processors and is interfaced with another Quad-core arm cortex A-53 processor using PCIe and I2C/GPIO interface and LPDDR4 memory to execute state of the art mobile vision models such as MobileNet v2 at almost 400 fps, in a power efficient manner.
Table 2 shows the overall performance of the chip for different neural networks algorithms which are trained using Imagenet dataset with 100 classes. A systolic array claims several advantages such as simple and regular design, concurrency, and balancing computation with I/O. However, until now, there has been no commercially successful processor based on a systolic array. The TPU is the first and it is impressive, arguably the largest systolic array implemented or even conceived. Figure 8 shows the block diagram architecture of another recent Mobileeye’s (an Intel Company) EyeQ5 machine vision processor at the edge [14].

More than 27 car manufactures have chosen the chip for their assisted-driving technologies based on its performance as it is able to support fully autonomous (level 5) vehicles. The 7 nm FinFet processor can provide up to 25 DNN TOPS for a maximal power of 10 W. It features several VLIW SIMD vector microcode processors (VMP) to provide hardware support for operations common to computer vision applications. It also features a Programmable Macro Array (PMA) which was initially introduced in the EyeQ4 and now reaches its 2nd generation of implementation which enables computation density nearing that of fixed-function hardware accelerators without sacrificing programmability. Another 8 core CPU engine is available to handle some complex tasks such as object tracking using the built-in Lucas and kanade algorithm [16], where the image inversion step does not fit the SIMD architecture. Geeley Auto Group is planning to use two chip by end of 2021 to yield level 4 autonomous vehicle using seven long range and four close range cameras and deliver 360\degree surrounds view to enable a scalable feature bundle supporting highway hands-free, navigation-based highway-to-highway, arterial, and up to urban hands-free driving. The chip is the fruit of joint efforts between Mobileye Mobileye and STMicroelectronics (STM) Leveraging Mobileye substantial experience in automotive-grade designs and STM support in state-of-the-art physical implementation, as well as automotive-grade memories, high-speed interfaces, and system-in-package design.

IBM company is one of the rare companies that suggest ASIC machine chips for asynchronous event-based sensors namely “silicon retinae” which are inspired by biological systems. It very recently suggested the 28 nm IBM’s TrueNorth all digital ASIC chip which performs about 46 Billion synaptic operations per second with 70 mW power consumption using 1 million neuros and 256 million synapses implemented using 5.4 billion transistors (Figure 9). The chip which comprises around 4096 cores (Each core comprises 256 input axons, 256 output neurons, and 256 x 256 synapses) was recently successfully used in real-time low power vision applications. It is the outcome of a decade research efforts under the DAPRA SyNAPSE program. TrueNorth chips can be connected directly together to form larger systems, and a circuit board with 16 chips has been developed (e.g. IBM’s NS16e board [9]). Nevertheless, one significant disadvantage of this chip is the weak data representation of neural networks trained with floating point precision, since the all communication between neuosynaptic cores are represented by binary spikes. This drop of accuracy is compensated using stochastic computing methods where the data is statically represented in both temporal and spatial domains using multiple spikes of the neural networks [10].
D. FPGA-based IoT and edge machine vision systems

Initially, FPGAs comprised large number of reprogrammable LUT which are interconnected together via local and global interconnection networks to implement various sequential and combinatorial logic circuits which can run in parallel. Two major companies, namely Altera and Xilinx, emerged for successfully providing different chips and associated software development tools. The high latency to implement large bit width arithmetic operations required in machine vision algorithms, in addition to the memory access bottleneck drove the manufactures to enhance the architecture of their respective chips by adding hundreds of DSP blocks and local onchip memories to implement various parallel architectures such as MIMD, SIMD architectures, and systolic architectures which are suitable for machine vision algorithms [12]. Nevertheless, this still remain not enough for DNN algorithms because of their low capacity to host all the weights which are required in DNN which involves intensive extensive external memory-FPGA data transfer increasing significantly the algorithm latency. Altera was recently acquired by Intel, which may indicate the future direction of next generation multicore CPU to incorporate reprogrammable devices to accelerate machine vision and image processing tasks. This is the case of the recent Xilinx Zynq Ultrascale + MPSoC FPGA which is one of the largest IC comprising million logic cells and include a quad-core 1.5 GHz 64 bits Cortex A-3 processor, and a Dual core Arm Cortex R5F making it suitable to host Xilinx’s deep learning processing unit (DPU), created for researchers working in hardwired machine learning algorithms [30]. The chip which supports several CNNs such as VGG, ResNet, GoogLeNet, YOLO, SSD, MobileNet, and FPN is supported by an associated software development kit which allows to perform pruning and quantization to satisfy low latency (DECENT), maps the neural network to the DPU instructions (DNCC), and handles resource allocation and DPU scheduling (N²Cube).

Very recently, in Arm TechCon exhibition, the chip was demonstrated to successfully perform traffic light detection [10]. When running CNN tasks, it achieves 14 images/sec/watt, which outperforms the Tesla K40 GPU (4 images/sec/watt). Also, for object tracking tasks, it reaches 60 fps in a live 1080p video stream. In [35], it was demonstrated that a similar FPGA, Intel’s Stratix-10, could yield 50% greater throughput over Titan X GPU using 6-bit fixed point data, while the GPU processor performs better for FP32 operands. Intel also offers a similar edge platform, namely Mustang-F100 [26], which is based on Intel’s FPGA Arria-10 FPGA, which according to the Company is more suitable than the 8 VPU-based Mustang-V100 platform in case additional applications such as Speech recognition are required. Otherwise for solely video processing applications Mustang-F100 platform is recommended. The FPGA chip contains 1,367 variable precision DSP block, 2,736 (18 x 19) multiplier, and 480,000 Logic Elements (LE) which are enough to host several DNN algorithms. This yields 1.5 TFLOPs with a power consumption of less than 60 W. In addition, the chip can yield a deterministic real-time image processing without buffering (as low as < 3 ms latency) and Inference up to 25 images/sec/W on Cafee running Alexnet. Very recently, Fujitsu is providing its PCIe 3.0 x8 deep learning accelerator board, AI 400 DL based on Arria 10 Gx 1150 FPGA chip and comprising 8 GB of DDR4-2133 memory [7].

V. EMBEDDED IoT and EDGE HARDWARE VISION SYSTEMS

Taking advantage of the excellent performance of IoT and edge processors dedicated for machine vision applications, various IoT and edge machine vision systems using either GPU, multicore CPU, ASIC, or FPGA are now available for a wide range of applications ranging from face detection to
perception systems for autonomous vehicles. For instance, NVIDIA has very recently released an embedded off the grid DNN camera which is based on its Jetson TX2 processor [38] to support Caffe, Tensorflow, and cuDNN DNN algorithms in real-time. However, its cost is relatively expensive for a typical IoT devices (e.g. 2,418 US$), which may prevent its dense deployment in public places. The camera which can be considered as one of the existing highest standards yields all kind of communication interfaces such as Gb Ethernet, WiFi 802.11, and Bluetooth and consumes around 7 to 15 w with 12-42 V for solar or industrial power. Wahtari Neural camera is another Linux-based camera, dedicated for real-time machine vision applications using a single intel’s Movidus Myriad-X VPU processor [9]. The camera can deliver inference at 45 frames and detect more than 7000 license plates/hour using just 18 W of power. The camera can host simultaneously three different neural networks. Gorilla is another company that provides edge machine vision cameras using Intel’s multiCPU Atom 2 GHz E3950 quadcore processor [10]. A similar binocular camera, iDS-2CD8426G0/F-I, using a not disclosed GPU processor is supplied by Hivision Digital technology to perform face recognition at the edge for up to five simultaneous input video streams (120 x 1080 pixels) [12]. The camera which uses mixed DNN and stereovision-based algorithms is dedicated to operate in public places to perform additional tasks such as perimeter intrusions, crowd density calculation as well as fall and loitering detection. Duha Technology is another company that provide a DSP-based camera system using Sonny image sensor technology that can handle up to 10,000 different face images and manage five image libraries [122].

A. Autonomous navigation in moving vehicles

Recently, with the vast improvements in computing technologies, e.g., sensors, computer vision, machine learning, hardware acceleration, and the wide deployment of communication mechanisms, e.g., Dedicated short-range communications (DSRC), Cellular Vehicle-to-Everything (C-V2X), and 5G, autonomous driving research has attracted massive attention from both the academic and automotive communities [13][14]. Many automotive companies have made enormous investments in this domain, including ArgoAI, Audi, Baidu, Cruise, Mercedes-Benz, Tesla, Uber, and Waymo, to name a few [15]. Nevertheless, the cost is one of the essential factors that affect the broad deployment of autonomous vehicles. According to [41], the cost of a level 4 autonomous driving vehicle attains 300,000 dollars, in which the sensors, computing device, and communication device cost almost 200,000 dollars. Another fact to pay attention to is that currently, the field-testing of level 2 autonomous driving vehicles mostly happens in places with good weather and light traffic conditions like Arizona and Florida. The real traffic environment is too complicated for the current autonomous driving systems to understand and handle easily. The objectives of levels 4 and 5 autonomous driving require colossal improvement of the computing systems for autonomous vehicles. Light company offers a low-cost multi-cameras array of different sizes to build the 3D map of the surrounding environment using stereovision technique [43]. The camera which comprises several lenses with different focal lengths is dedicated for autonomous vision cars to substitute the high cost and lower range LIDAR systems. The stereovision algorithm is implemented using a non-disclosed FPGA to yield 95 millions points/s, which according to the company will be substituted by an ASIC chip [24]. This follows other research works where FPGA-based stereovision algorithms were successfully implemented into FPGA-based board, such as the 2W Xilinx’s Spartan 6-based FPGA real-time stereo camera [25]. This outperforms the Mobiley’s single camera which uses one single camera to rely on some inferencing to estimate the distance to a target by performing image segmentation and object detection for measuring the object size in pixels, which is rather a very hit and miss system. Even multi-camera systems used by companies like Tesla are challenged because they use three lenses of varying focal length clustered together. MobileEye company is also offering another very recent vision system for autonomous cars which is based on the property EyeQ5 processor (two processors in their recent system) [26], interfaced with Intel’s Atom 3xx4 CPU to yield 60% more performance at the same 30W power consumption as NVIDIA’s new Jetson Xavier chip. The system yields hands-free high driving, automated parking, in addition to other legacy autonomous driving features.

B. Face recognition and detection for IoT and edge cameras

Several cameras have been suggested for face recognition, and most of them include a pair of stereovision cameras to estimate the 3D position of the face to ensure an easy and accurate detection of the face within the scene. For instance, in [37], a 1.2 GHz 4 core A5 CPU together with ARM’s MALI 400 MP GPU processor were used for a 2 Mpixels binocular face recognition system to recognize up to 20,000 different faces within a database of up to 200,000 faces for a distance ranging from 0.5 to 3 m within a latency of less than 0.5 s, which is still excessively high for some real-time applications. The accuracy of the system is claimed to be 99.9 %. Aaeon Company is another supplier which provides an edge solution for face recognition using Gigabit Ethernet and the NVIDIA’s Jetson Tx2 processor [27]. The board has four Intel LAB (i211) ports to support 4 simultaneous IP cameras at the edge.

VI. PENDING CHALLENGES and PROSPECTS

As was mentioned throughout this paper, one of the pending challenges with IoT and edge-based machine vision devices remain the excessive power consumption of mainly the processors. As can be shown in Figure 10, FPGA offers the best speed-up per Watt in terms of DNN algorithm. However, GPU offers the highest speed-up but at the expense of higher power consumption. Being power hungry, GPU are challenged by other alternatives such as ASIC technology where Google’s TPU [29], Graphcore IPU [34] and amzon Inferential [35] are just few examples. To tackle
This paper summarizes the main recent findings in hardware architecture dedicated for IoT and edge devices. Four different technologies for processor design are emerging: GPU, multicore CPU, FPGA, ASIC or a combination of some of them, which is the present trend. Despite unprecedented growth in processing abilities of these processors for machine vision applications, the actual performance is bounded by the low bandwidth, relatively high power consumption, and high latency of data movement between the processors and memory subsystems, especially if they are located in different chips. While FPGA offers the best performance per Watt, GPU is more adequate for applications requiring very high speed-up with relaxed constraint on the power consumption, they require some hardware skills and usually time-consuming low-level programming to successfully implement a given application. A top-down and down-top hardware-software co-design approach by simultaneously designing the hardware platform and the machine vision algorithm can help optimizing the hardware solution in terms of area, power, and latency. This approach is more obvious for FPGA platform because of its configurability at the fine grain level which can yield an immense optimization using a proper high-level synthesis tool. Thus, there is a further need to explore more aggressively some features of machine vision algorithms such as data redundancy which is abundant in both image processing algorithms and DNN inferring models, as well as providing a dynamic quantization of their weights at different layers. Giant semiconductor companies are continuously increasing their revenues in AI-based accelerators, such as the actual 3.5 Billion US$ AI-related revenue for Intel [16]. Intel and Nvidia are engaged in battle over AI chips for autonomous vehicles with their state-of-the-art processors EyeQ5 and Jetson Xavier AGX respectively. This will bring great opportunities to researchers to build their IoT and edge machine vision chips for other applications. Nevertheless, one major issue facing scientists for using the state of the art commercially available AI accelerators is the lack of detailed documentation that allows full exploration of the chip. These include EyeQ5 and Jetson Xavier AGX processors, as well as other chips from car manufacturers Waymo, Uber and GM Cruise. Disclosing the datasheets and other associated software documentation of these processors to researchers would help achieving great machine vision systems.

VII. CONCLUSION

Table 3 below compare the GPU and FPGA implementations for some image processing algorithms. It can be noticed that while FPGA outperforms GPU for small kernel image convolution algorithms, NVIDIA’s performs better in terms of stereovision [16]. Very recently, IBM has licensed what it claims to be the smallest and most powerful processor using 2 nm technologies which can yield 45% higher performance and about 75% lower energy usage than today’s most advanced 7-nm chips [28]. The chip is expected to be produced starting year 2024. This will pave the way to more accelerated machine vision and AI algorithms. In efforts to let programmers taking full advantage of the complex hardware architecture of FPGAs, several researchers suggested hardware compilers and high-level synthesis (HLS) tools to directly map DNN algorithms into FPGA platforms. Example of such tools is the DNNWeaver which seamlessly generate a synthesizable hardware accelerator for a given (DNN, FPGA) pair from high level specification in Caffe [28]. In [29] researchers from Intel revealed that using this tool for either Altera’s Stratix V and Arria-10 or Xilinx’s Zynq FPGAs, a speed-up over most existing powerful multicore CPU (i.e. Xeon E3) and many-core GPUs (Tegra K1, GTX 650 Ti and Tesla K40) could be achieved. ARM A15 CPU core was also assessed to perform the weakest with the advantage of very low power.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
Photogrammetry, Remote Sensing & Spatial Information Sciences, vol. 42, no. 4, 2018.

[6] N. T. Karim, S. Jain, J. Moonrinta, M. N. Dailey, and M. Ekpanyongpang, “Customer and target individual face analysis for retail analytics,” in 2018 International Workshop on Advanced Image Technology (IWAIT), IEEE, 2018, pp. 1–4.

[7] A. Hajirassolhiah and Co, “Suitability of recent hardware accelerators (DSPs, FPGAs, and GPUs for computer vision and image processing algorithms”, Signal Processing: Image Communication Vol. 68, pp. 101–119, 2018.

[8] D. Bhowmik and K. Appiah, “Embedded vision systems: A review of the literature,” in Proc. Int. Symp. Appl. Reconfigurable Comput., Santorini, Greece: Springer, 2018, pp. 204–216.

[9] M. T. Banday, “A study of current trends in the design of processors for the internet of things,” in Proceedings of the 2nd International Conference on Future Networks and Distributed Systems. ACM, p. 21, 2018.

[10] “Edge AI Inference Rugged System with 6th Gen Intel® Core i5-6442EQ QC and Two Intel MA2485 VPUs”, available in URL: anewtech.net/sites/default/files/pdf/Anewtech-AI-inference-system-AD-AIR-200.pdf. Last visited in 4/1/2021.

[11] M. Meribout and M. Motomura, “A combined approach to high-level synthesis for dynamically reconfigurable systems”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 14(6), pp. 1508-1522, 1995.

[12] KM Hou and Co “A reconfigurable and flexible parallel 3d vision system for a mobile robot”, 1993 Computer Architectures for Machine Perception, 215-221, 1993.

[13] M Meribout, M. Nakanish and T. Ogura, “Hough transform implementation on a reconfigurable highly parallel architecture”, Proceedings Fourth IEEE International Workshop on Computer Architectures for Machine Perception. CAMP'97, pp. 186-194, 1997.

[14] M. P. Jao and Co, “High performance embedded computing”, in Embedded Computing for High Performance, 2017.

[15] M Meribout, T. Ogura, and M. Nakanishi. On using the CAM concept for parametric curve extraction. IEEE Transactions on Image Processing, 9 (12), 2126-2130, 2000.

[16] M Meribout, M. Nakanish and T. Ogura, “A parallel algorithm for real-time object recognition”, Pattern recognition 35 (9), 1977-1931, 2002.

[17] “GAP8 Hardware Reference Manual”, Available at URL: https://www.amazon.com/dp/B077LZP4WQ. Last visited on 3/29/2021.

[18] C. Foster et al, “SVO: fast semi-direct monocular visual odometry”, IEEE International conference on Robotics and Automation ICRA, 2013.

[19] “IBM says it has created the world’s smallest and most powerful microchip”, URL: https://www.3dnews.ru/2017/05/06/tech/ibm-semiconductor-two-nanometer/index.html. Last accessed on 5/7/2021.

[20] “IP rating model”, URL: https://www.mpi.ca/index.php?controller=DealershipMain&action=show&controllerId=12. Last accessed on 3/23/2021.

[21] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, “Deepdriving: Learning affordance for direct perception in autonomous driving,” in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2722–2730.

[22] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang et al., “End to end learning framework for autonomous driving,” arXiv preprint arXiv:1604.07316, 2016.

[23] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, “Deep reinforcement learning framework for autonomous driving.” Electronic Imaging, vol. 2017, no. 19, pp. 70–76, 2017.

[24] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, “Learning to drive in a day,” in 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 8248–8254.

[25] U. Muller, J. Ben, E. Casas, B. Flepp, and Y. L. Cun, “Off-road obstacle avoidance through end-to-end learning,” in Advances in neural information processing systems, 2006, pp. 739–746.

[26] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, “An open approach to autonomous vehicles,” IEEE Micro, vol. 35, no. 6, pp. 60–68, 2015.

[27] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt et al, “Towards fully autonomous driving,” In Conference on Neural Information Processing Systems, 2011. IEEE Intelligent Vehicles Symposium (IV). IEEE, 2011, pp. 163–168.

[28] C. Urmsen, and Co, “Autonomous driving in urban environments:Boss and the urban challenge,” Journal of Field Robotics, vol. 25, no. 8, pp. 425–466, 2008.

[29] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, “A survey of autonomous driving: Common practices and emerging technologies,” IEEE Access, vol. PP, pp. 1–13, 2020.

[30] O. Mano and Co, “Forward Collision Warning with a Single Camera”, in Proc. of the IEEE Intelligent Vehicles Symposium (IV2004), June 2004, Parma, Italy.

[31] G. P. Stein, O. Mano and A. Shashua, “Vision-based ACC with a Single Camera: Bounds on Range and Range Rate Accuracy”, in Proc. of the IEEE Intelligent Vehicles Symposium (IV2003).June 2003, Columbus, OH.

[32] “Will Camera Startup Light Give Autonomous Vehicles Better Vision than Lidar?”, IEEE Spectrum. Article available online in URL: https://spectrum.ieee.org/transportation/sensors/will-camera-startup-light-give-autonomous-vehicles-better-vision-than-lidar..

[33] Christian Szegedy, Sergey Ioe, Vincent Vanhoucke, and Alexander A. Alemi. 2017. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In Association for the Advancement of Artificial Intelligence.

[34] Chenyi Chen, Ari Se, Alain Kornhauser, and Jianxiong Xiao, “Deepdriving: Learning A,ordance for Direct Perception in Autonomous Driving.” in IEEE International Conference on Computer Vision.

[35] Dhiraj Gandhi, Lerrel Pinto, and Abhinav Gupta, “Learning to Fly by Crashing”, In IEEE/RSJ International Conference on Intelligent Robots and System.

[36] Jianqiu Qiu et al. Going deeper with embedded FPGA platform for convolutional neural network on FPGA. 2016.

[37] X. Zhang et al. High-performance video content recognition with long-term recurrent convolutional network for FPGA. In FPL, 2017.]Junsong Wang et al. Design flow of accelerating hybrid extremely low bit-width neural network in embedded FPGA. In FPL, 2018.

[38] S. Han and Co. Ese: Efficient speech recognition engine with sparse LSTM on FPGA. In FPL, 2017.

[39] “An end-to-end open source machine learning platform”, https://www.tensorflow.org/ Last visited on 9/23/2021.

[40] “TensortFlow”, URURI https://nge.nvidia.com/catalog/containers/nvidia/tensorflow. Last visited on 23/09/2021 and TFLite (tensortflow Lite) from Google [”"] URL: https://www.tensorflow.org/lite. Last visited on 23/09/2021.

[41] B. Hassibi and Co., “Second Order Derivatives for Network Pruning: Optimal Brain Surgeon. In Conference on Neural Information Processing Systems, 1993.

[42] Xiaofan Zhang et al. DNNBuilder: an automated tool for building high-performance DNN hardware accelerators for FPGAs. In ICCAD, 2018.

[43] A. Alemi. 2017. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In Association for the Advancement of Artificial Intelligence.

[44] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. MobileNet: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

[45] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenets2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4510–4520, 2018.

Mahmoud Meribout (M’91) received the B.Eng. degree in 1985 and the Ph.D. degree in electronics engineering from the University of Technology of Compiègne, Compiègne, France, in January 1995. He was with Nippon Telegraph and Telephone Corporation, Tokyo, Japan, from 1995 to 2000, where he was involved in several projects related to embedded systems design. In 1998, he received the NTT Best Award for his research and development efforts in the areas of embedded systems design and imaging systems. In 2008, he joined the Electrical Engineering Department, Petroleum Institute, where he is currently a full Professor. His current research interest focusses on embedded systems, imaging systems (THz, MPI, optical, electrical, and magnetic tomography), instrumentation, and multiphase flow metering.
### Table 1: NVIDIA Jetson Processors performance (in fps) for different DNN models

| Model                        | Jetson Nano | Jetson TX2 series | Jetson Xavier NX | Jeston AGX |
|------------------------------|-------------|-------------------|------------------|------------|
| Inception V4 (299 x 299)     | 13          | 32                | 405              | 704        |
| VGG-19 (224 x 224)           | 12          | 29                | 313              | 432        |
| Super Resolution (481 x 321) | 15          | 33                | 166              | 302        |
| Unet (256 x 256)             | 17          | 39                | 166              | 251        |
| OpenPose (256 x 456)         | 15          | 35                | 271              | 484        |
| Tiny Yolo V3 (416 x 416)     | 49          | 112               | 618              | 1127       |
| ResNet-50 (224 x 224)        | 47          | 112               | 1100             | 2109       |
| SSD Mobilenet-V1 (300 x 300) | 48          | 109               | 1058             | 1919       |
| SSD ResNet34 (1200 x 1200)   | 1           | 2                 | 29               | 55         |
| BERT_BASE (seq-Length = 128) | -           | -                 | 115              | 286        |
| BERT_LARGE (seq-Length = 128)| -           | -                 | 35               | 90         |

### Table 2: Time per inference (ms) for different model architectures and image sizes

| Model architecture     | Desktop CPU¹ | Desktop CPU² + USB accelerator (USB 3.0) | Embedded CPU² | Dev Board³ with edge TPU |
|------------------------|--------------|------------------------------------------|---------------|--------------------------|
| Unet Mv2 (128 x 128)   | 27.7         | 3.3                                      | 190.7         | 5.7                      |
| DeepLab V3 (512 x 513) | 394          | 52                                       | 1139          | 241                      |
| DenseNet (224 x 224)   | 380          | 20                                       | 1032          | 25                       |
| Inception v1 (224 x 224)| 90          | 3.4                                      | 392           | 4.1                      |
| Inception v4 (299 x 299)| 700         | 85                                       | 3157          | 102                      |
| Inception-ResNet V2 (299 x 299) | 753 | 57                                      | 2852          | 69                       |
| MobileNet v1 (224 x 224)| 53          | 2.4                                      | 164           | 2.4                      |
| MobileNet v2 (224 x 224)| 51          | 2.6                                      | 122           | 2.6                      |
| MobileNet v1 SSD (224 x 224)| 109 | 6.5                                     | 353           | 11                       |

¹ Desktop CPU: Single 64-bit Intel(R) Xeon(R) Gold 6154 CPU @ 3.00GHz
² Embedded CPU: Quad-core Cortex-A53 @ 1.5GHz
³ Dev Board: Quad-core Cortex-A53 @ 1.5GHz + Edge TPU
Table 3: FPGA and GPU performance for different image processing algorithms [27]

| Algorithm                        | FPGA                                      | GPU                                   |
|----------------------------------|-------------------------------------------|---------------------------------------|
| Parallel Convolution             | NVIDIA Tesla C1060 outperformed for large kernels | Xilinx’s iRetex-5 outperformed for small kernels |
| 3D Ultrasound computer tomography| Processing time was 21.86 ms using Viretex-7 FPGA | Nvidia’s Tesla K20 performed 1.86 slower than Viretex-7 FPGA |
| 2D Convolution                   | 125 fps for an image size of 1280 x 720 and kernel size of 25 | 80 fps for an image size of 1280 x 720 and kernel size of 25 |
| Disparity measurement            | Xilinx Virtex-6: 6 M disparities/s        | Nvidia GTX280: 7.2 M disparities/s     |
| Image classification using MobileNet v1, ResNet v, and Inception | -                                        | Jetson TX2 yielded 7.52 % improvement in testing accuracy and 1.8 x faster speed than other single DL model |
| Image classification using MobileNet v1, ResNet v, and Inception | -                                        | Nexus 5x yielded 3x speedup of matrix multiplication compared to Tensorflow |