# 1.2 Intelligence on Silicon: From Deep-Neural-Network Accelerators to Brain Mimicking AI-SoCs

**Hoi-Jun Yoo,** *KAIST, Daejeon, Korea*

## 1. Introduction

While, currently, Artificial-Intelligence technology is affecting all industrial paradigms, it is also impacting lifestyle of all society. AI technology is widely used in most information hardware, software, and networking, underlying all consumer technology: smartphones, home appliances, and the Web.

Figure 1.2.1 shows 3 main application areas for AI technology: Cloud, Edge, and Mobile. The first is Cloud AI where intelligent applications are processed by server machines at data centers with results sent to edge controllers (such as speakers) and through to mobile devices (such as smartphones). As AI applications become more popular, servers need to incorporate dedicated AI accelerators to speed up the ever-increasing demand for AI processing with low energy consumption [1, 2].

In addition to AI processing in servers, recently many edge devices such as CCTV or AI speakers (as depicted in Figure 1.2.1) are required to process AI algorithms directly themselves, as so-called "on-device AI" [3]. This is needed, for example, in the case of an edge CCTV camera since most of the data which it collects and sends to the server is mundane, within infrequent critical events. Thus, it is apparent that sending all of the useless data to the server is a waste of network communications and server computation power. Accordingly, in this case, if the edge device has intelligent functions to extract only the meaningful data and send only the selected data to the server, the communication traffic can be reduced with much quicker response from the server. But, the need to minimize communication and latency extends beyond the Edge to the mobile devices, themselves. Correspondingly, mobile devices (such as smartphones, automobiles, and drones) will perform AI functions with their own processors avoiding round-trip access to the Edge or server. For these applications, real-time operations are required to avoid obstacles or collision with fast decision making with local intelligence even when they are moving at high speed. This is especially critical for smartphones which increasingly process high quality AI functions, such as face recognition and gesture recognition for user authentication, or natural user interface (UI), all with ultra-low power consumption. Smartphones are of particular concern because their security function is always-on.

In order to satisfy AI processing demands, high-performance silicon processors have already been developed to provide AI functions to the servers [1, 2, 4]. Currently, convolutional neural network (CNN) algorithms are accelerated using dedicated parallel processing elements (PE) to enhance convolution matrix multiplication, and to reduce external memory access. More recently, other deep-neural-network algorithms, such as recurrent neural network (RNN) and Restricted Boltzmann Machine (RBM) have been realized as accelerator chips [5]. As well, many DNN accelerators have been introduced as NPU (neural processing unit) and DPU (deep learning processing unit) or silicon IPs, taking the hardware intensive CNN operations from the main CPU to reduce the computation burden on the main processor [6].

However, the requirements for edge and mobile DNN accelerators are quite different from DNN accelerators used for server applications. Since many users want to run different AI functions and AI algorithms in the server, the server accelerator should support various types of DNN algorithms with high precision and large external memory. In contrast, the mobile accelerator usually supports only a limited DNN configuration and an algorithm for a specific application in real-time, with low power consumption, reduced number of precision, and limited memory size.

From an applications point of view, there will be (at least) two types of AI systems in the future: The first of these is an AI system replacing the human in the control loop (as in current Advanced Driver Assistance System (ADAS) applications) shown in Figure 1.2.2(a). Most current AI chips are targeting this category aiming to precisely control the machines just as well (or even better than) a human would. The second kind of AI system exhibits humanistic intelligence to collaborate symbiotically with human beings as illustrated in Figure 1.2.2(b). It structures the AI system to be more social, more like a pet or a human being, rather than like a machine which is unaffected by its user's emotions. In this situation, the AI system needs sensing ability to facilitate interaction with human users intelligently and emotionally, such as face recognition, emotion recognition, gesture recognition, and communication in natural language. (More details about such human-like intelligence will be explained in Section 3-C.)These social abilities are required particularly in mobile devices, and in contrast with AI in autonomous control, should be continuously active since its time of need is not predictable. Thus, low power consumption with high performance is critical, and energy efficiency rather than simple performance, is the most important parameter. In the following sections, the details of possible chip implementation schemes will be presented.

In this paper, we will examine the recent trends demonstrated in actual chip implementations of CMOS DNN accelerators; this is intended to present a sharp contrast with conventional mere descriptions of proposed unimplemented DNN chip architectures [7, 8, 9]. In Section 2, real chip implementation is emphasized for mobile AI applications using real chip DNN accelerators designed for high energy efficiency and high flexibility. In Section 3, the need for local learning will be introduced with new learning algorithms and hardware architectures. In particular, deep reinforcement learning will be emphasized as the next major application area for local learning. As well, humanistic intelligence will be introduced as a way to enhance the interaction between AI systems and human beings. In Section 4, we will describe a brain-mimicking AI SoC using two different approaches: One applies neuromorphic computing employing new non-volatile memories and mixed-mode circuits to mimic the neuron level activities of the brain. The second employs the connectome brain map to directly mimic functions of parts of the brain, and their interconnections. And finally, in Section 5 conclusion will be drawn.

## 2. CMOS DNN Accelerator ASICs

There have been many proposals for DNN accelerators [6] and many hardware architectures. As well, FPGA optimization schemes have been developed for DNN accelerators for high performance inference [6, 10]. At the same time, DNN algorithms are still evolving: A few software-centric works (such as SqueezeNet, MobileNets, and ShuffleNet) have been proposed to reduce the number precision without accuracy loss allowing reduced on-chip memory size to enable the DNN operations on existing mobile CPUs [5, 6, 10, 11, 12, 13]. For example, MobileNets can be installed on mobile devices because it is as accurate as VGG16 [14] while being 32 times smaller and 27 times less compute intensive. But the application of these software-centric techniques is very limited and performance degrades unbearably if they are applied to different applications from the optimized one. In this respect, FPGAs may be the way to utilize the most recent advanced algorithms, but with excess power consumption and limited performance compared to that of a custom ASIC. Recently, DNN accelerator architectures incorporate the flexibility of dynamically reconfigurable hardware architectures to meet the changing requirements of the new algorithms. It is important to note that in addition to the acceleration provided by the DNN algorithm, it also allows for greater energy efficiency. In the remainder of Section 2, the trends of the real chip implementation issues will be examined with emphasis on the energy efficiency and the reconfigurable architectures.

### 2.1 Energy Efficiency and Accuracy

DNN accelerators are optimized for convolutional computation with on-chip parallel processors, which also integrate the on-chip memory to store the weights and images, reducing access to the external memory. High-performance DNN accelerators have been developed for server applications and are currently successfully employed [1, 2, 4, 10]. However, the available approaches cannot obtain high energy efficiency (computation capability per unit power). Energy efficiency is particularly important in mobile DNN processors, along with the high computation capability of DNN algorithms. In order to enhance the energy efficiency without accuracy loss, not only the hardware but also the algorithms themselves should be optimized. The frequently used co-optimization schemes are Data Reuse, Reduced Number Precision, Weight Quantization, Sparsity Exploitation, Tensor Decomposition, Pruning, Approximate Computation, and Teacher-Net Learning [6]. These techniques can reduce the power consumption, chip complexity, on-chip memory size, and the off-chip memory access, but there may be a trade-off between the inference accuracy and chip complexity.

In addition to the DNN configuration and chip architecture level optimizations, circuit level optimizations are also useful for real chip implementation to reduce the power consumption and increase computation speed [15]. For example, the complicated multiplier can be replaced with a Look-up Table with quantization schemes to increase energy efficiency [16]. As well, the cell circuitry of the on-chip SRAM can be customized to speed up the vertical cell access [17], and sub-threshold logic can be used to enhance energy efficiency [17]. Potentially, an NVM (Non-Volatile Memory) rather than SRAM can enhance the energy efficiency further, as will be discussed in Section 3.

 978-1-5386-8531-0/19/$31.00 

Figure 1.2.3 shows the performance-inference power efficiency of the reported DNN chips. 10TOPS/W is the criterion of high energy efficiency, and only the ASIC approach can meet the criterion. In addition to the above hardware-neural network co-optimization schemes, adopting small weight widths of 1bit and 4bits can lead to more than 10TOPS/W efficiency. Reducing the bit length to binary weights (BinaryConnect) and binary inputs (BNNs) will reduce the complexity of chips and reduce computation time, but accuracy degradation is unavoidable as a result [18]. Therefore, the programmability of bit lengths is required to secure the accuracy in addition to energy efficiency.

Figure 1.2.4 illustrates the trade-off between programmability and energy efficiency. Even though a GPU is not so versatile as a CPU, it is more effective in graphics-intensive applications, and an NPU, less programmable than a GPU, is more efficient in a convolution-intensive application. Since the programmability of an NPU is restricted compared to that of a GPU, there is concern that it may not accommodate new DNN algorithms in accordance with the evolution of algorithms. Recently, the flexibility of DNN accelerators has been enhanced with reconfigurable hardware, making it possible to accommodate various DNN algorithms on a single accelerator chip.

## 2.2 High Flexibility

Figure 1.2.5(a) shows 8 reported DNN chips as a function of energy efficiency and programmability. Programmability or hardware flexibility is as important as energy efficiency in DNN accelerators for optimizing mobile applications. A flexible computing architecture can customize the hardware architecture to provide exactly the required DNN processing capability needed. Therefore, it provides the programmability of hardware, just like FPGAs do for DNN accelerators while speeding up the processing time by an order-of-magnitude over fixed-instruction-set CPU with far less power consumption. There are two approaches to increase flexibility of DNN accelerators. The first is to use heterogeneous architectures, while the second is to adopt reconfigurable architectures.

Two parameters can be utilized to enhance the flexibility of DNN processors, the neural network hardware architecture and bit precision length with either coarse grain or fine grain as shown in Figure 1.2.5(b). Our first example is to integrate diverse types of DNNs, in this case, CNN and RNN, together in coarse grain [19]. Since the CNN has different optimal architecture from the RNN hardware, heterogeneous accelerators are designed for CNN and RNN separately. In addition, RNN and fully-connected layer (FCL) can share the same hardware architecture, RNN hardware can be reconfigured as the FCL. With software programming, the chip can be reconfigured to select the DNN architecture such as a CNN only or CNN+FCL or CNN+FCL+RNN [19]. The second approach is to reconfigure the bit precision in coarse grain with fine-grain architecture or the architecture in coarse grain with fine-grain bit precision [20]. For the given neural network, the bit precision of the weights and feature maps can vary 1 to 4bits layer by layer to reduce the required computation without significant impact on accuracy. In addition, both the DNN architectures and the bit length can be reconfigured together. For example, coarse bit length selection between 8bits and 16bits for PE functions, together with fine-grain hardware reconfiguration (such as the number of PE allocation) can be performed with software according to neural network types (CNN, RNN, and FCL), and it can be extended to use binary/ternary weights [21]. The third example is the most-advanced reconfigurable architecture, for which details are shown in Figure 1.2.6. Figure 1.2.6(a) shows both DNN architecture and the bit precision to be reconfigured dynamically in fine grain to further enhance the flexibility and the energy efficiency of the DNN accelerators [22]. Figure 1.2.6(b) explains its bit-serial architecture to vary the bit precision continuously from 1bit to 16bits. The network architecture can be changed dynamically as the fine-grain reconfigurable processing (DRP) even layer by layer to make it more versatile.

## 3. Local Learning and Intelligence

Learning is regarded as the basis of intelligence. Currently, learning for DNNs (that is weight determination) is done remotely, typically in the cloud server and downloaded to edge/mobile devices. It is a popular concept that learning is so computationally intense that only a high-performance server can perform it properly using available big data. However, the demand for local learning is increasing and a hardware solution for it is required.

## 3.1 Edge and Mobile Learning

Currently, DNN learning demonstrably takes a long time; For example, ResNet50 employs 1.2M images for learning, taking 11 days for training using a single Nvidia P40 GPU. Many DNN accelerators have been developed to speed up the training time in the cloud server [1, 2, 4]. A server-centric system as shown in Figure 1.2.7 uploads a large amount of data from mobile devices to the cloud server through a long wide area network, resulting in uncontrolled latency and high data-transmission energy consumption. Also, privacy may not be protected because even private data must be sent to the cloud server to use the intelligent applications. In addition, run-time training is required to filter out malicious or inconsistent data to protect the security of the system from adversarial attacks. Moreover, training of an edge or mobile device is necessary to realize the quick adaptation to the new environment, while reducing energy consumed by communications and the cloud server. For example, a pruned network can be re-trained on a mobile device with mobile learning for personalization and robustness [23]. Also, federated learning is proposed to protect privacy by keeping the sensitive data on the mobile devices while the learning task is solved by a loose federation of participating mobile devices which are coordinated by a central server [24].

Recently, local learning accelerators have been proposed for real-time object tracking and low-power local adaptation. A new algorithm, Feedback Alignment, is used to integrate on-line learning and inference on a single chip, making on-line learning possible in a mobile environment for real-time accurate object tracking [25]. Figure 1.2.7(b) provides an example of the datapath of a local CNN learning-accelerator chip. Weights are stored in a column buffer (C-buf), and feature maps are broadcasted to PEs and error maps are moved to the PEs from output memory to generate weight gradients. The weight gradients are aggregated temporally on accumulation registers of each PE, and then the results are stored to C-buf. As shown in the figure, the weight-gradient data path is reversed compared to that of the inference. It utilizes data sparsity optimally, together with variable number precision, enabling local mobile CNN learning at high speed and with low power [26].

## 3.2 Deep Reinforcement Learning

Reinforcement learning (RL), as illustrated in Figure 1.2.8, enables the AI system to find its optimal next operation in the unknown environment by repetitive self-learning. RL needs a reward to learn its optimal operation. Sometimes, the reward is clear, such as when a robot falls when it steps on irregular terrain. Of course, sometimes, in the real world, the reward cannot be clearly defined. In this case, the reward can be predicted in three possible ways: use demonstrated human behaviors as the reward (imitation learning, inverse reinforcement learning (IRL)); decide on the reward by itself (self-supervised learning, model-based learning); or obtain the reward with the help of other modalities.

The policy function of DQL (Deep Q-Learning) uses DNN for on-chip learning which enables the AI system with DQL to update its policy function. AI robot and the autonomous car will utilize Deep RL to make decisions much as a human being does, in a new environment different from the one in which it learned. In this case, remote learning may suffer from unpredictable latency, and local learning is mandatory to obtain secured RL with on-site data in real-time, such as for car-driving as illustrated in Figure 1.2.8. The Figure illustrates a multi-step process of DRL with four functions: the AI system in the car observes its state using its sensors; the policy function is implemented as the DNN; the DNN receives the state (how fast the car moves, whether there are pedestrians, the street conditions, and so on) as its input and generates the next action; and the AI system calculates the reward as the feedback from the resulting output action. Following a few thousand action cycles and the collection of the observed states, the policy-update process starts to modify the weights in the DNN. With the newly modified policy, the action is regenerated and the policy update is repeated. Usually, this process requires massive computation and memory size, requiring a full function DRL chip with RNN learning or FC-layer learning as reported in [27].

## 3.3 Humanistic Intelligence

As explained in Figure 1.2.2, AI systems and human beings will collaborate in the near future, on the basis of mutual understanding. The required trust and understanding are acquired by the AI system through local learning and recognition of the intent and emotion of the user. In this case, the AI system can be regarded as a pet or a companion with humanistic intelligence so that users can communicate or even share their emotions with the AI system. Learning

techniques, such as DRL, will be the key technology for the personalization and collaboration of the AI system. But, as well, the human user can also learn to adapt their behavior over time. For example, the ever-changing emotions of the user can act as a reward for the DRL process, with the robot learning and modifying its behavior by recognizing the user's emotions [28]. Figure 1.2.9 provides examples of an emotion recognition DNN accelerator [22], as well as a gesture recognition accelerator based on CNN and RNN [19]. Human gestures, such as clapping illustrated on the left side of the Figure, must be deciphered according to the temporal sequence of human poses. The RNN receives the output of the CNN to recognize the gesture. The emotion can be detected based on the shapes of the facial muscles using CNN. The right side of the Figure shows an example of emotion recognition (choosing one of 7 emotion categories), using a chip for facial frontalization and CNN [29].

## 4. Brain-Mimicking AI SoCs

There are two different ways to mimic the brain functions as shown in Figure 1.2.10. The first is the neuromorphic approach which tries to mimic the neuronal behavior, and the second is the functional modeling approach which mimics the cognitive behavior of a part or all of the brain.

### 4.1 Neuromorphic Approach

The neuromorphic approach of Figure 1.2.10 can be divided again into two methods, synapse centric and neuron centric. The first one is to mimic the basic "synaptic" behavior such as spike neural network or pulsed neural network with analog-digital mixed-mode circuits. Information can be conveyed by spike timing, for example, using the interval between spiking signals (such as spike-timing-dependent plasticity (STDP)), providing large information transmission capacity [30, 31]. SRAMs are used to store the digital values of the weights which are converted into analog, and multiplied with the input pulses to obtain output spiking signals. The design consumes less power than current Von Neumann architectures, but as a novel technique, it lacks high performance AI functions due to their less developed AI algorithms and limited integration scale due to SRAM cell size (100 to 200$F^2$, 10 times more than DRAM NVM cells).

If the synapse can be realized with a single device rather than a complicated circuit, the integration level can be easily scaled up with reduced power consumption. The non-volatile memories with 4$F^2$ to 12$F^2$ cell size such as Memristor (using Resistive RAM (RRAM)), Phase Change RAM (PCRAM), and Magnetic RAM (MRAM) have been actively studied as synaptic devices. If such a device can be integrated with neuronal circuits, non Von Neumann architectures such as biological neuronal computation can be realized, minimizing power consumption needed to transfer weight values from memory to the synapse [32, 33, 34]. Furthermore, in principle, the Memristor can also store analog weights so that a compact analog spiking neural network could be easily realized. Such Processor in Memory (PIM) (or Computation in Memory (CIM)) technology is a promising direction for the realization of AI functions. However, such cell-device technology is not yet mature, and only limited array sizes have been realized in actual silicon for small applications including character recognition or low-resolution face recognition [32].

The neuron-centric approach is to create an analog circuit which mimics the behavior of the "neuronal" cell faithfully [35, 36, 37, 38, 39, 40, 41]. Their attempts of reproducing biological synaptic or neuronal behaviors have resulted in CMOS circuits, requiring more than 10 transistors. Correspondingly, their integration though providing a faithful reproduction of neuronal behavior, is limited by the large number of transistors per cell. However, using this scheme, new AI-supporting functions such as low-energy event-detection sensors can be developed with a limited number of neurons [42].

### 4.2 Functional-Model Approach

The functional-model approach mimics the behavior of a part of the brain (or of the whole brain). Cognitive psychology studies the computational models of human behavior, along with the advances in anatomy show that networks of the brain (connectomes) play an essential role in brain intelligence [43]. Thus the hippocampus can be mimicked as the key to "Artificial General Intelligence" [44]. In addition, brain functions such as "episodic memory" can be modeled to speed up AI learning mimicking the past-experience memory [45]. Even the whole brain functions can be modeled as a hierarchical combination of multiple machine learning functions each mimicking a functional module of the brain [46]

Another interesting brain-mimicking example concerns "visual attention". Our eyes receive a huge stream of visual data ($10^8$ to $10^9$ bits/second). Nevertheless, its processing is possible because the brain's visual attention function confines visual data for possible high-level cognitive processing, such as object recognition or scene interpretation. Visual attention has been modeled from psychological, neurobiological, and computational perspectives, and especially, from the anatomy and functionality of the human visual system front end. This explains how, when, and why, human select behaviorally relevant image regions. A saliency map can be realized using a neural network based on the Treisman's "Feature Integration Theory" [47], with its computational architectures already developed for possible applications [48, 49]. Based on this theoretical study, single and multiple visual attention has been implemented on silicon, as shown in Figure 1.2.12 [50, 51]. There are two types of attention, bottom-up and top-down. Bottom-up attention is mainly based on characteristics of a visual scene (stimulus-driven) whereas top-down attention (goal-driven) is related to cognitive phenomena such as memory, expectations, reward, and current goals [52]. Another important type of attention illustrated in Figure 12 is spatio-temporal attention which depends on the changes due to egocentric movements or dynamics of the surrounding world [53]. Recently, concepts of visual attention, such as spatial [54], channel-wise [55], and zoom-in [56], have been applied to CNNs to improve performance for object classification. Furthermore, residual connections are added to various attention algorithms to improve accuracy [57, 58].

Brain-mimicking AI can be realized, not simply with machine-learning modules (including DNNs) to reproduce the local functional blocks of the brain, but also using cognitive architecture based on the neurobiological connectome and knowledge from the cognitive behavioral experiments.

## 5. Conclusions

Most of DNN researches have been associated with software solutions, with recent proposals for improved DNN processors. However, these designs are inappropriate for mobile applications and often unsuited for silicon-chip implementations. As well, for human-computer interaction, social intelligence measures (such as emotion recognition and gesture recognition) must be available with low-power consumption.

For true progress in AI, DNNs and their hardware architectures must be examined from the point-of-view of silicon-chip implementation, especially, for edge and mobile applications. In such applications, energy efficiency is very important, and a holistic optimization must be undertaken involving, neural network architectures, hardware architectures, and even circuits. In addition, to overcome deficiencies of high energy efficiency silicon DNN, the reconfigurability of hardware is necessary. For this, both neural network hardware architecture and bit-precision must be re-configurable at both fine grain and coarse grain levels, for more versatility and higher energy efficiency.

Learning is the most important attribute of AI. Until now, all learning is done primarily on high-powered servers at remote data centers. But, as energy costs and latency force intelligent functions to move to the edge and mobile devices (as "on-device AI"), local learning provides personalization and privacy of learning functions. Moreover, reinforcement learning will be used for fine tuning of intelligent functions needed for local and very detail data. The humanistic intelligence factors such as emotion sharing and intent recognition also require local learning, and low power, making high-performance learning processors mandatory.

The brain can be mimicked either structurally (as an architecture) or functionally (behaviorally). Some neuromorphic approaches try to mimic the spiking signals of synapses and neurons. Such a synapse can be realized with memory, typically SRAM. However, in order to reduce area and power consumption, non-volatile memories such as RRAM (Memristor), PC-RAM, and MRAM are being actively studied. Neuronal behavior can be modeled with analog CMOS circuits using as many different pulse shapes as needed for biological neurons for information transfer and processing. Some designs try simply to understand biological behavior of the brain based on neuromorphic VLSI, while others are used simply for a particular application such as in low-energy event-detection camera. Another interesting approach is to mimic the functionality of the brain based on a computational model combining knowledge of cognitive science and "connectome". An interesting and active topic is the visual attention of the brain. The computational model of the brain's visual attention is used to select only the salient region of the image to speed up processing with the limited computational resources of the chip. Bottom-up attention, top-down attention, and spatio-temporal attentions have been explained.

 

The brain continues to inspire our search for Machine Intelligence. Correspondingly, AI and DNN research will likely make more progress by utilizing brain-mimicking.

## References

[1] N. P. Jouppi et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit," ISCA, Vol. 17 1–12, 2017.
[2] NVIDIA Corporation, "Tesla V100 Data Center GPU," 2017. Available at: https://www.nvidia.com/en-us/data-center/tesla-v100/
[3] MIT Technology Review, "On-Device Processing and AI Go Hand-in-Hand," 2018. Available at: https://www.technologyreview.com/s/610421/on-device-processing-and-ai-go-hand-in-hand/
[4] S. M. Tam et al., "SkyLake-SP: A 14nm 28-Core Xeon® Processor," ISSCC, pp. 34-36, 2018.
[5] X. Zhang et al., "Shufflenet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," arXiv, 1707.01083, 2017.
[6] V. Sze et al., Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," Proc. of the IEEE, Vol. 105, No. 12, pp. 2295-2329, 2017.
[7] Y. Chen et al., "DaDianNao: A Machine-Learning Supercomputer," MICRO, pp. 609-622, 2014.
[8] A. Parashar et al., "SCNN: An Accelerator for Compressed-sparse Convolutional Neural Networks," ISCA, Vol. 45, No. 2, pp. 27-40, 2017.
[9] S. Venkataramani et al., "SCALEDEEP: A Scalable Compute Architecture for Learning and Evaluating Deep Networks," ISCA, pp. 13-26, 2017.
[10] S. Han, "Accelerating Inference at the Edge," Hotchips tutorial, 2018
[11] A. Howard et al., "Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv, 1704.04861, 2017.
[12] M. Sandler et al., "Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation," arXiv, 1801.04381, 2018.
[13] F. N. Iandola et al., "Squeezenet: Alexnet-Level Accuracy with 50x Fewer Parameters and <0.5MB Model Size," arXiv, 1602.07360, 2016.
[14] K. Simonyan et al., "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv, 1409.1556, 2015.
[15] H. J. Yoo, "Mobile/Embedded DNN and AI SoCs," Symp. on VLSI Technology and Circuits, Short Couse, June 2017.
[16] J. Lee et al., "A 21mW Low-Power Recurrent Neural Network Accelerator with Quantization Tables for Embedded Deep Learning Applications," A-SSCC, pp. 237-240, 2017.
[17] K. Bong et al., "A 0.62mW Ultra-Low-Power Convolutional-Neural-Network Face-Recognition Processor and a CIS Integrated with Always-On Haar-Like face Detector," ISSCC, pp. 248-249, 2017.
[18] M. Courbariaux et al., "Binaryconnect: Training Deep Neural Networks with Binary Weights During Propagations," arXiv, 1511.00363, 2015.
[19] D. Shin et al., "DNPU: An 8.1TOPS/W Reconfigurable CNN-RNN Processor for General-Purpose Deep Neural Networks," ISSCC, pp. 240-241, 2017.
[20] S. Yin et al., "A 1.06-to-5.09 TOPS/W Reconfigurable Hybrid-Neural-Network Processor for Deep Learning Applications," Symp. on VLSI Circuits, pp. 26-27, 2017.
[21] K. Ueyoshi et al, "QUEST: A 7.49 TOPS multi-purpose log-quantized inference engine stacked on 96 MB 3D SRAM using inductive coupling technology in 40 nm CMOS", ISSCC, pp.216-218, 2018.
[22] J. Lee et al., "UNPU: A 50.6TOPS/W Unified Deep Neural Network Accelerator with 1b-to-16b Fully-Variable Weight Bit-Precision," ISSCC, pp. 218–219, 2018.
[23] P. S. Chandakkar, Y. Li, P. L. K. Ding and B. Li, "Strategies for Re-Training a Pruned Neural Network in an Edge Computing Paradigm," IEEE EDGE, pp. 244-247, 2017.
[24] J. Konecny et al., "Federated learning: Strategies for Improving Communication Efficiency," arXiv, 1610.05492, 2016.
[25] D. Han et al., "A 141.4 mW Low-Power Online Deep Neural Network Training Processor for Real-time Object Tracking in Mobile Devices," ISCAS, pp. 1-5, 2018.
[26] J. Lee et al., " LNPU: A 25.3TFLOPS/W Sparse Deep-Neural-Network Learning Processor with Fine-Grained Mixed Precision of FP8-FP16," ISSCC, Paper7.7, 2019.
[27] C. Kim et al., "A 2.1 TFLOPS/W Mobile Deep RL Accelerator with Transposable PE Array and Experience Compression," ISSCC, Paper 7.4, 2019
[28] iMotions, A. S. (2015). Affectiva iMotions Biometric Research Platform. from https://imotions.com/affectiva/
[29] S. Kang et al, "B-Face: 0.2 mW CNN-Based Face Recognition Processor with Face Alignment for Mobile User Identification" Symp. On VLSI Circuits, pp. 137-138, 2018.
[30] F. Akopyan et al., "TrueNorth: Design and Tool Flow of a 65mW 1 Million Neuron Programmable Neurosynaptic Chip," IEEE TCAD, Vol. 34, No. 10, pp. 1537–1557, 2015.
[31] M. Davies et al., "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning," IEEE Micro, Vol. 38, No. 1, pp. 82-99, 2018.
[32] P. Yao et al., "Face Classification Using Electronic Synapses," Nature Communications, Vol. 8, p. 15199, 2017.
[33] S. Park et al., "Electronic System with Memristive Synapses for Pattern Recognition," Scientific reports, Vol. 5, p. 10 123, 2015.
[34] W. Chen et al., "A 65nm 1Mb Nonvolatile Computing-in-Memory ReRAM Macro with Sub-16ns Multiply-and-Accumulate for Binary DNN AI Edge Processors," ISSCC, pp. 494-496, 2018.
[35] S. Choudhary et al., "Silicon Neurons that Compute," ICANN, pp. 121–128, 2012.
[36] G. Indiveri et al., "ReRAM-Based Neuromorphic Computing," Resistive Switching: From Fundamentals of Nanoionic Redox Processes to Memristive Device Applications, pp. 715-730, 2016.
[37] R. Serrano-Gotarredona et al., "CAVIAR: A 45k Neuron, 5M Synapse, 12G Connects/s AER Hardware Sensory–Processing–Learning–Actuating System for High-Speed Visual Object Recognition and Tracking," IEEE Trans. on Neural Networks, Vol. 20, No. 9, pp. 1417-1438, 2009.
[38] S. B. Furber et al., "The SpiNNaker Project," Proc. of the IEEE, Vol. 102, No. 5, pp. 652-665, 2014.
[39] S. Joshi et al., "Scalable Event Routing in Hierarchical Neural Array Architecture with Global Synaptic Connectivity," IEEE CNNA, pp. 1-6, 2010.
[40] B. V. Benjamin et al., "Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations," Proc. of the IEEE, Vol. 102, No. 5, pp. 699-716, 2014.
[41] FACETS project, "Fast Analog Computing with Emergent Transient States," Available at: http://www.facets-project.org
[42] S. C. Liu et al., "Event-Based Neuromorphic Systems," New York, NY, USA: Wiley, 2014.
[43] H. S. Seung, "Reading the Book of Memory: Sparse Sampling Versus Dense Mapping of Connectomes," Neuron 62 (1), 17-29, 2009.
[44] K. L. Stachenfeld et al., "The Hippocampus as a Predictive Map," Nature Neuroscience, Vol. 20, No. 11, .pp. 1643, 2017.
[45] D. Hassabis et al., "Neuroscience-Inspired Artificial Intelligence," Neuron, Vol. 95, No. 2, pp. 245–258, 2017.
[46] N. Arakawa et al., "The Whole Brain Architecture Initiative," ICONIP, pp. 316-323, 2016.
[47] S. Lee et al., "Familiarity Based Unified Visual Attention Model for Fast and Robust Object Recognition," Pattern Recognition, Vol. 43, pp. 1116–1128, 2010.
[48] A. Treisman, G. Gelade,"A Feature-Integration Theory of Attention," Cognitive Psychology, Vol. 12, No. 1, pp. 97–136, 1980.
[49] L. Itti et al. "A Model of Saliency Based Visual Attention for Rapid Scene Analysis," IEEE TPAMI, 20(11):1254–1259, 1998.
[50] K. Kim et al., "A 125GOPS 583mW Network-on-Chip Based Parallel Processor with Bio-inspired Visual-Attention Engine," ISSCC, pp. 308-615, 2008.
[51] J. Kim et al., "An Attention Controlled Multi-Core Architecture for Energy Efficient Object Recognition," Signal Processing: Image Communication, Vol. 25, No. 5, pp. 363-376, 2010.
[52] S. Lee et al., "A 345mW Heterogeneous Many-Core Processor with an Intelligent Inference Engine for Robust Object Recognition," IEEE JSSC, Vol. 46, No. 1, pp. 42–51, 2011.
[53] J. Oh et al., "A 320mW 342GOPS Real-Time Moving Object Recognition Processor for HD 720p Video Streams," ISSCC, pp. 220–224, 2012.
[54] M. Jaderberg et al., "Spatial Transformer Networks," NIPS, pp. 2017-2025, 2015.
[55] B. Zhou et al., "Learning Deep Features for Discriminative Localization," CVPR, pp. 2921-2929, 2016.
[56] J. Fu et al., "Look Closer to See Better: Recurrent Attention Convolutional Neural Network for Fine-Grained Image Recognition," CVPR, pp. 4476-4484, 2017.
[57] F. Wang et al., "Residual Attention Network for Image Classification," CVPR, pp. 6450-6458, 2017.
[58] L. Chen et al., "SCA-CNN: Spatial and Channel-Wise Attention in Convolutional Networks for Image Captioning," CVPR, pp. 6298-6306, 2017.

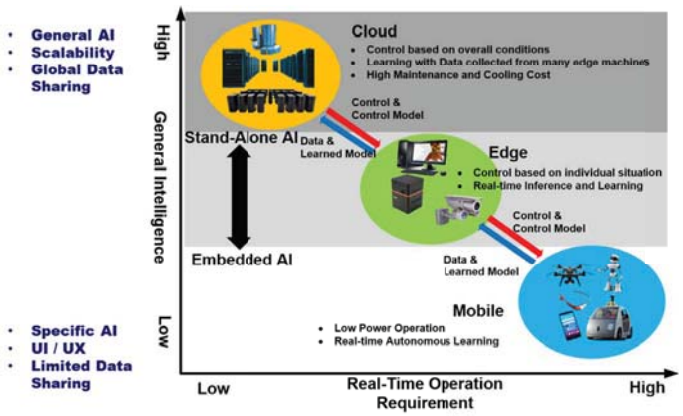


Figure 1.2.1: AI Applications.

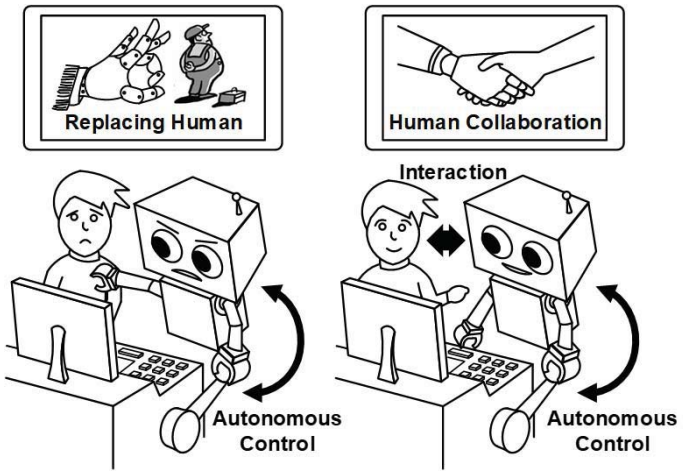


Figure 1.2.2: Application Types of AI systems.

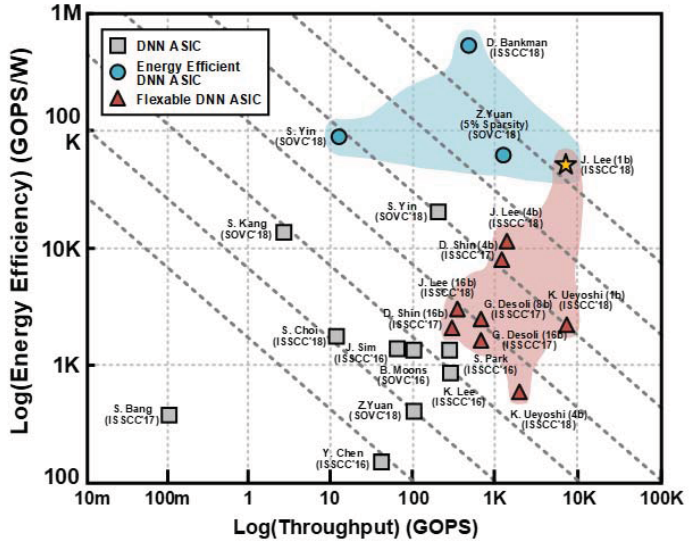


Figure 1.2.3: Energy efficiency of reported DNN chips.

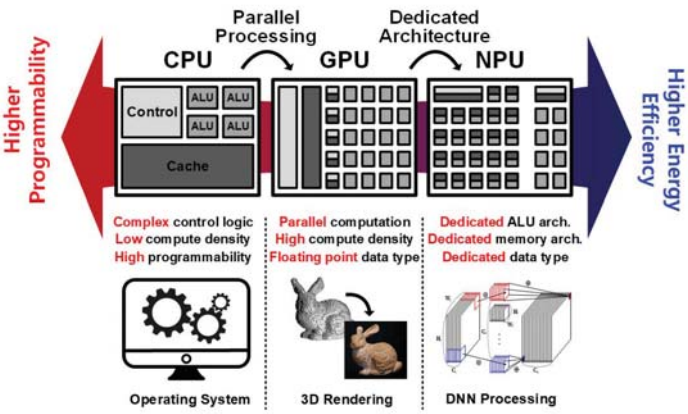


Figure 1.2.4: Comparison of processors: CPU, GPU, and NPU.

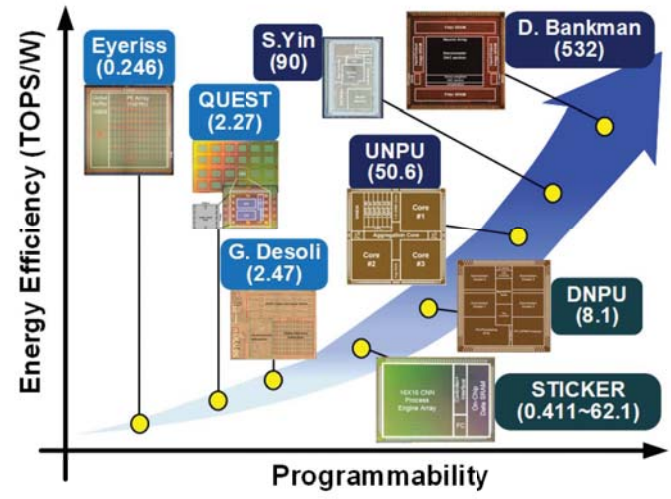


Figure 1.2.5(a): Energy Efficiency and Programmability of DNN Solutions.

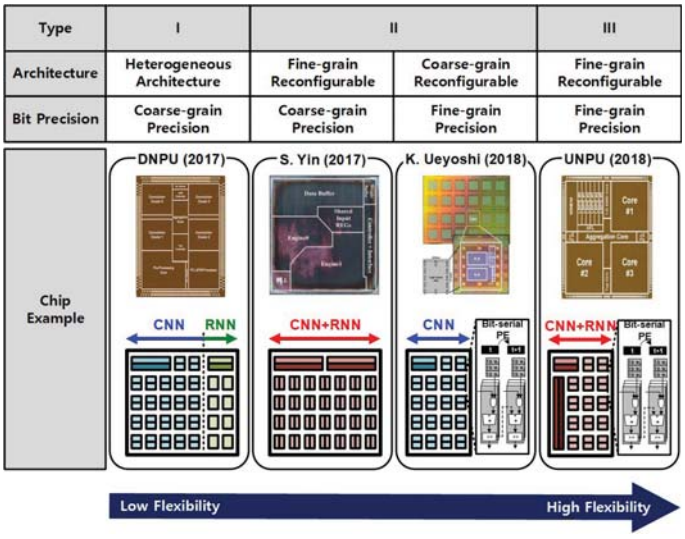


Figure 1.2.5(b): Reconfigurability and Precision variation of DNN Solutions.

 978-1-5386-8531-0/19/$31.00 

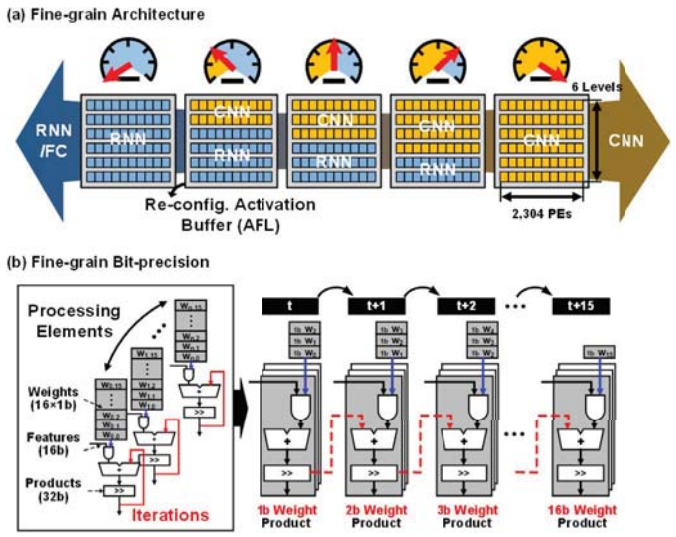


Figure 1.2.6: Fine-Grain Dynamic Reconfigurable DNN Processor with Fine-Grain Precision Variation.

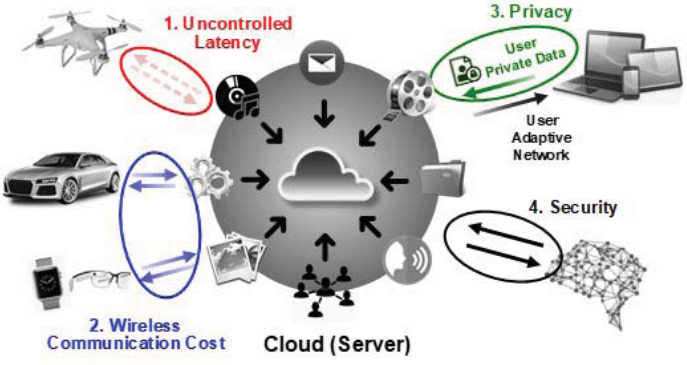


Figure 1.2.7(a): Server-Centered DNN Training Problems.

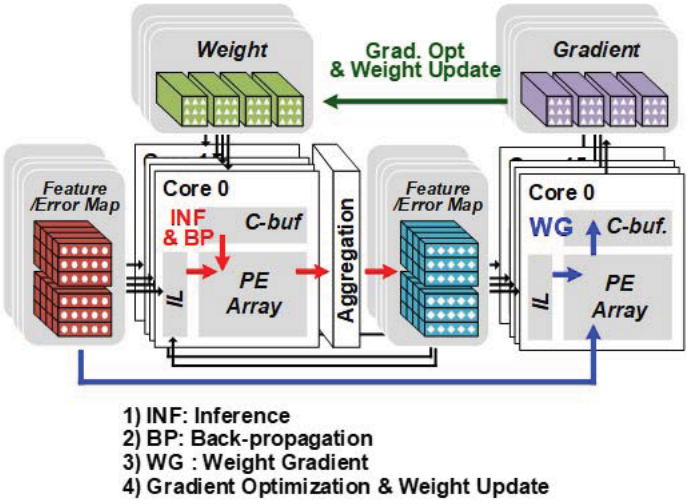


Figure 1.2.7(b): Architecture for On-Chip Learning.

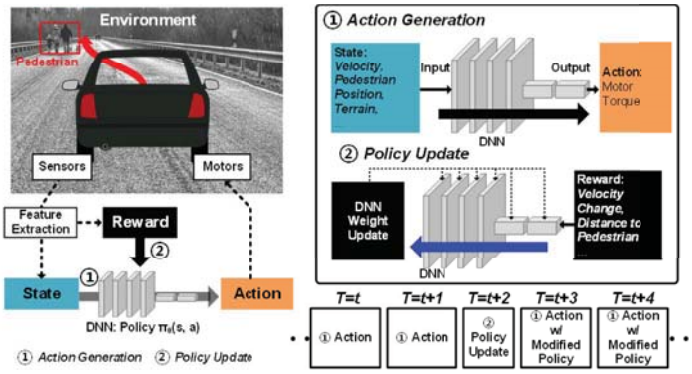


Figure 1.2.8: Online Learning for Deep Reinforcement Learning.

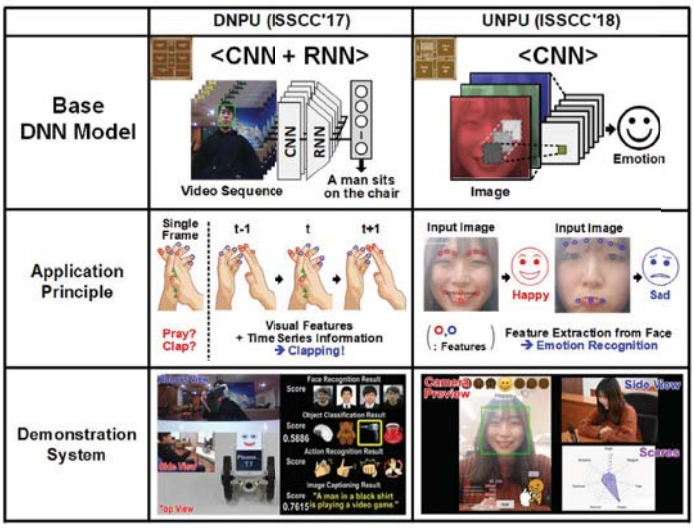


Figure 1.2.9: Gesture Recognition and Emotion Recognition.

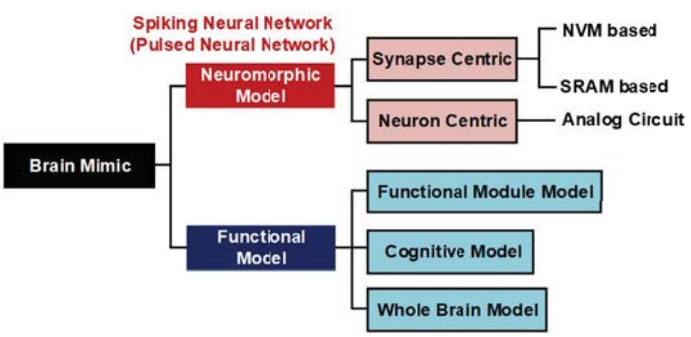


Figure 1.2.10: Types of Brain-Mimicking Approaches.

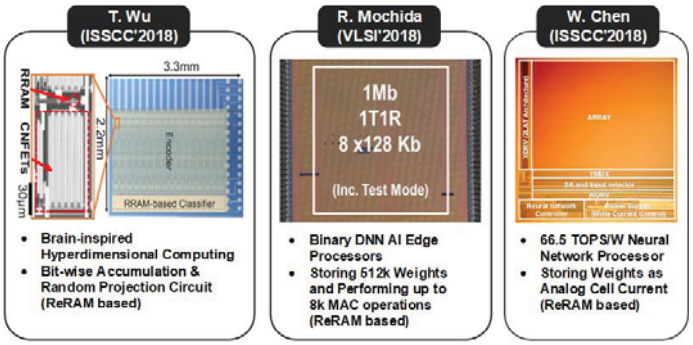


Figure 1.2.11: DNN Accelerator based on NVM PIM architectures.

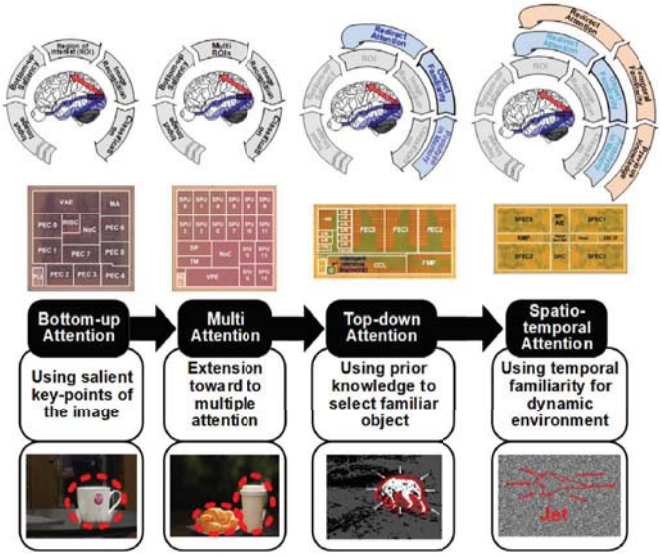


Figure 1.2.12: AI-SoCs based on Brain-Mimicking Visual Attention.



 978-1-5386-8531-0/19/$31.00 ©2019 IEEE