Understanding the Landscape of Accelerators for Vision

Nandhini Chandramoorthy,
Karthik Swaminathan, Matthew
Cotter, Xueqing Li,
Vijaykrishnan Narayanan
Pennsylvania State University

Indranil Palit, Sharon Hu,
Michael Niemier
University of Notre Dame

Kevin Irick
Silicon Scapes Inc.

***Abstract*—** Visual analytics applications are becoming ubiquitous and embedded in various systems that we interact with daily. Limited power budgets and the need for high performance for cognitive visual analytics have led to a three-pronged approach of integrating advances in algorithms, architectures and technology towards designing next generation vision accelerators. Vision applications benefit from increasing processor customization, emerging devices and technologies such as Tunnel-FETs and Resistive-RAMs, and trends in non-Boolean computing such as Cellular Neural Networks (CNNs) and neuromorphic architectures. This paper provides an overview of the evolving landscape of vision accelerators.

Keywords
Emerging devices; Heterogeneous architecture; Non-Boolean computation;

I. INTRODUCTION

Visual perception, image processing and analytics applications have become pervasive across automotive, medical, retail, education, agriculture, personal and security domains. Algorithms of increasing computational complexity are used in these diverse domains to provide enhanced user experience. For example, facial recognition algorithms can be used in retail stores to identify customers and provide a personalized shopping experience. Feature points from edge detectors and depth information are used in automatic parking assist systems. Video surveillance systems rely on object detection and tracking algorithms. Figure 1 shows a typical object recognition pipeline and sample images showing application of these algorithms in vehicle navigation assistance, personal analytics and surveillance.

Implementation of such ‘smart’ applications on hardware poses severe computational challenges in today’s energy-limited processors. There have been concerted efforts in exploiting new architectures, algorithms and emerging device technologies to achieve high performance, energy efficient processing. Most image processing tasks demonstrate common computational characteristics which can be well-exploited by a synergistic approach in device, architectural and algorithmic design domains. Research efforts in System-on-Chip architecture have focused on processor customization in order to improve performance and energy efficiency. Emerging technologies such as Tunnel FETs, Resistive RAMs and Spin Transfer Torque (STT) RAMs are being proposed as complements, or even as alternatives to existing logic and memory technologies, on account of their superior characteristics such as power efficiency and density. Recent work in the analog computing domain propose models for power efficient computation of low level image processing tasks. Non-Boolean computation for image processing such as Cellular Neural Networks (CNNs) implemented using Tunnel-FETs and symmetrical-graphene-insulator-graphene FETs (SymFETs) offer significant advantages over traditional architectural approaches. Research trends in each of these design spaces exploit characteristics of smart vision tasks to efficiently map different classes of applications on these platforms. Figure 2 shows a taxonomy of image processing and analytics systems illustrating design approaches in conventional CMOS architectures, emerging devices and technologies, CMOS analog computing and non-Boolean computing.

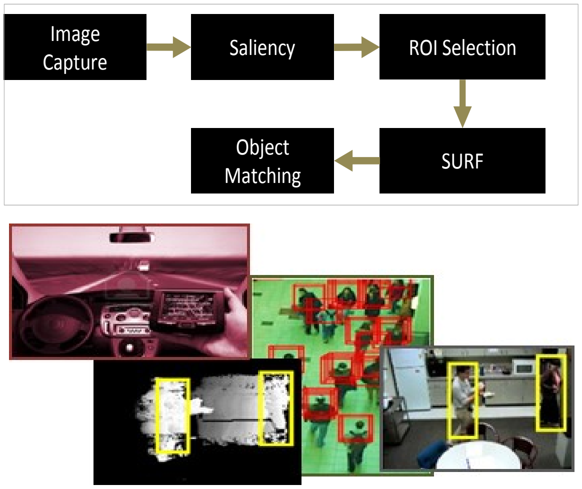


Figure 1: Sample object recognition pipeline including image capture, visual saliency map computation, SURF object recognition and matching. Images show application of recognition and tracking algorithms in different domains

In this work, we explore trends in such heterogeneous architectures and examine the design features that make them amenable for efficient implementation of image processing applications. We also compare the implementations of

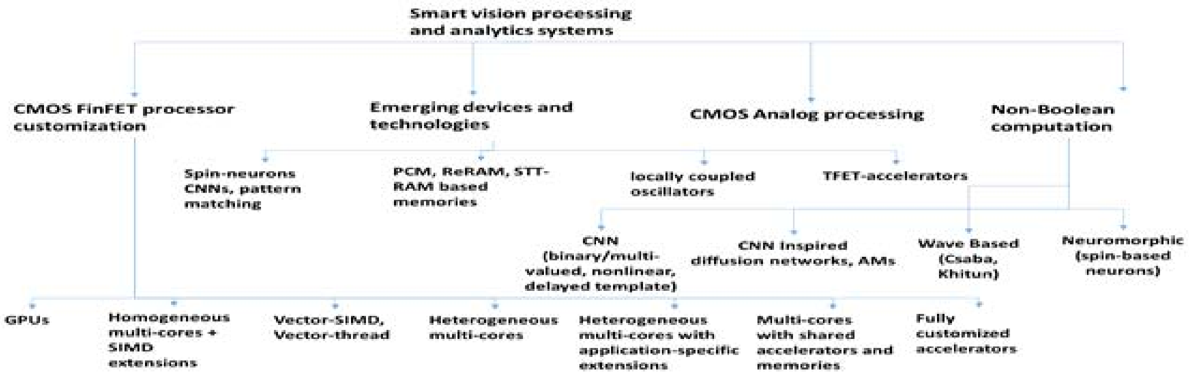

Smart vision processing and analytics systems
CMOS FinFET processor customization
Emerging devices and technologies
CMOS Analog processing
Non-Boolean computation
Spin-neurons CNNs, pattern matching
PCM, ReRAM, STT-RAM based memories
locally coupled oscillators
TFET-accelerators
CNN (binary/multi-valued, nonlinear, delayed template)
CNN inspired diffusion networks, AMs
Wave Based (Csaba, Khitun)
Neuromorphic (spin-based neurons)
GPUs
Homogeneous multi-cores + SIMD extensions
Vector-SIMD, Vector-thread
Heterogeneous multi-cores
Heterogeneous multi-cores with application-specific extensions
Multi-cores with shared accelerators and memories
Fully customized accelerators

Figure 2. A taxonomy of image processing systems showing trends in CMOS SoCs, emerging devices and non-Boolean computation

several algorithms on systems that comprise different points in the device, architecture and algorithm design space.

II. Customized CMOS Architectures

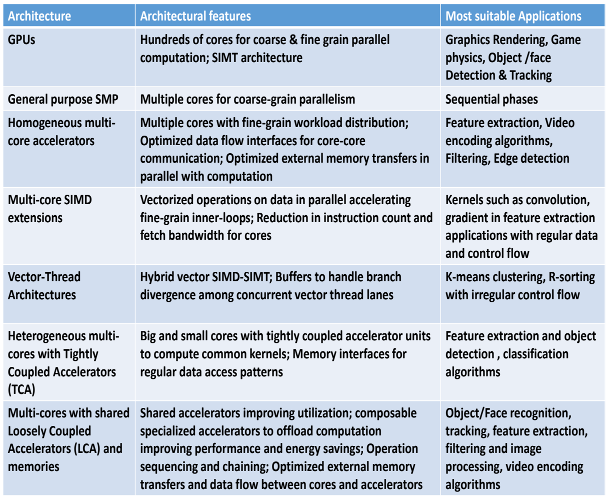
With the end of Dennardian scaling [1], recent industry and academic efforts in processor architecture have focused on processor customization as a solution to improve performance and energy efficiency, also taking advantage of rising transistor count with technology scaling. Processors such as Nvidia's Tegra 4, Samsung's Exynos 5, Octa and Tilera's Gx72 [2] employ heterogeneous cores and domain-specific accelerators for maximum performance and improved energy efficiencies for media-rich applications. Most computer vision and image processing applications have common characteristics such as fine-grain parallelism, coarse-grain parallelism, structured memory accesses, streaming data access patterns, and pipelined data flow. Most customized image processing and analytics systems exploit these characteristics to provide high performance and low energy processing. The micro-architectural features that allow efficient implementation of image processing applications are summarized in Figure 3a. In this section we describe several classes of heterogeneous architectures used in vision systems, as follows.

A. GPUs: GPUs offer hundreds of cores for computation and are well suited for embarrassingly parallel algorithms. Graphics and video processing and scene analysis applications can be partitioned into segments processed by independent thread blocks, individual threads within blocks processing smaller sub-regions in parallel using the CUDA programming model. The Single Instruction Multiple Thread (SIMT) architecture schedules thread blocks concurrently on available multiprocessors, hiding memory latencies using abundant thread-level parallelism.

B. Homogeneous multi-cores accelerators: Multi-core clusters such as Tilera Gx [2], Platform-2012 [3] make use of abundant data level parallelism present in tasks such as feature extraction. Platform2012 [3] proposes an array of CPUs with independent instruction streams and shared L1 memories with DMA engines. Each image is segmented into tiles and the workload is distributed among cores enabling parallel computation. Code segments shown in Figure 3b are good candidates for such multicore accelerators. Unlike SMPs with cache-coherency and memory consistency models, DMA engines in Platform-2012 transfer data from external memories into shared L1 memories overlapping computation on the cores. The Polymorphic Pipeline Array (PPA) [4] is a multi-core accelerator aimed at exploiting fine-grain and coarse-grain parallelism found in streaming applications. The PPA consists of a large number of simple cores each with multiple processing elements and shared scratchpad memories connected using a Mesh-style interconnect. Direct connections between register files in neighboring cores enable fast sharing and forwarding of data. Cores can also be combined logically to create a larger virtual core which can speed up inner-loop fine grain parallelism. The compiler converts application task graphs into instruction schedules using Virtualized Modulo Scheduling [4] while the hardware dynamically allocates resources. Run-time virtualization is possible by transferring instructions from a core's loop buffer to the neighboring core's loop buffer. Applications such as H.264, AAC video encoders with coarse-grain pipeline parallelism and inner-loop level parallelism show improvement in performance by exploiting fine-grain parallelism using modulo scheduling.

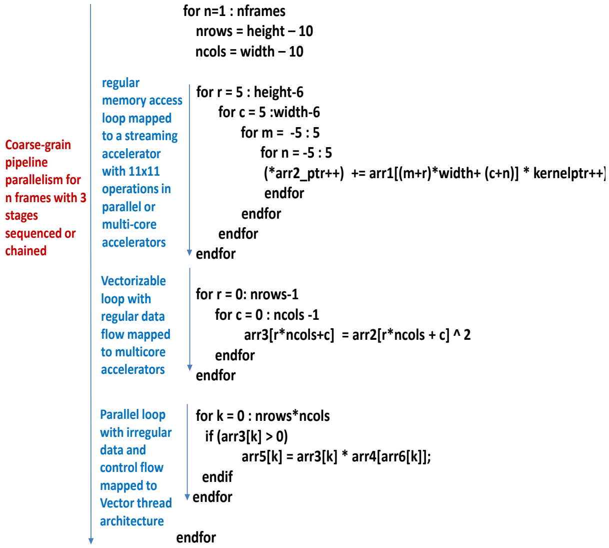
C. Homogeneous multi-cores with SIMD extensions: SIMD extensions such as ARM-Neon [5] save computation energy by operating on 128 bit vectors in multiple vector lanes of computing units in parallel, rather than fetch and process scalar instructions.. Neon instructions consist of vector load/store and compute instructions and also perform data copying between general purpose registers and vector registers. Compilers can autovectorize fine-grain loops exploiting data-level parallelism and reducing instruction fetches. Code segments with regular control flow such as the second one in Figure 3b are good examples.

D. Vector Architectures: Traditional vector processors map elements on multiple vector lanes or pipes of deeply pipelined functional units in a striped fashion, with data chaining between units. Special vector memory instructions perform strided or indexed memory accesses to load data elements from memory into vector registers. Lee et al propose a vector-thread architecture, Maven [6], a hybrid of vector-SIMD and SIMT architectures. The advantage of



Architecture	Architectural features	Most suitable Applications
GPUs	Hundreds of cores for coarse & fine grain parallel computation; SIMT architecture	Graphics Rendering, Game physics, Object /face Detection & Tracking
General purpose SMP	Multiple cores for coarse-grain parallelism	Sequential phases
Homogeneous multi-core accelerators	Multiple cores with fine-grain workload distribution; Optimized data flow interfaces for core-core communication; Optimized external memory transfers in parallel with computation	Feature extraction, Video encoding algorithms, Filtering, Edge detection
Multi-core SIMD extensions	Vectorized operations on data in parallel accelerating fine-grain inner-loops; Reduction in instruction count and fetch bandwidth for cores	Kernels such as convolution, gradient in feature extraction applications with regular data and control flow
Vector-Thread Architectures	Hybrid vector SIMD-SIMT; Buffers to handle branch divergence among concurrent vector thread lanes	K-means clustering, R-sorting with irregular control flow
Heterogeneous multi-cores with Tightly Coupled Accelerators (TCA)	Big and small cores with tightly coupled accelerator units to compute common kernels; Memory interfaces for regular data access patterns	Feature extraction and object detection , classification algorithms
Multi-cores with shared Loosely Coupled Accelerators (LCA) and memories	Shared accelerators improving utilization; composable specialized accelerators to offload computation improving performance and energy savings; Operation sequencing and chaining; Optimized external memory transfers and data flow between cores and accelerators	Object/Face recognition, tracking, feature extraction, filtering and image processing, video encoding algorithms

Figure 3a: Summary of architectures useful for image



```
for n=1 : nframes
  nrows = height – 10
  ncols = width – 10

  for r = 5 : height-6
     for c = 5 :width-6
        for m =  -5 : 5
           for n = -5 : 5
              (*arr2_ptr++)  += arr1[(m+r)*width+ (c+n)] * kernelptr++]
           endfor
        endfor
     endfor
  endfor

  for r = 0: nrows-1
     for c = 0 : ncols -1
        arr3[r*ncols+c]  = arr2[r*ncols + c] ^ 2
     endfor
  endfor

  for k = 0 : nrows*ncols
    if (arr3[k] > 0)
          arr5[k] = arr3[k] * arr4[arr6[k]];
    endif
  endfor

endfor
```

Coarse-grain pipeline parallelism for n frames with 3 stages sequenced or chained
regular memory access loop mapped to a streaming accelerator with 11x11 operations in parallel or multi-core accelerators
Vectorizable loop with regular data flow mapped to multicore accelerators
Parallel loop with irregular data and control flow mapped to Vector thread architecture

Figure 3b: Code characterization for heterogeneous mapping

Maven over vector-SIMD architectures lies in the handling of irregular control and data flow among vector threads such as in Figure 3b. Handling branch divergence among vector threads using flags can lead to complicated flag arithmetic logic for complex conditions. Maven proposes a SIMT-like solution, where the threads with taken branches are buffered while the others execute, followed by the divergent vector threads. Most classification and recognition-based algorithms such as k-means clustering and radix sort which have irregular control flows benefit from such architectures.

E. Heterogeneous multi-cores: Platforms such as *ARM* big.LITTLE [7] consist of high performance Cortex-A15 cores and low power/high efficiency Cortex-A7 cores connected using a cache-coherent interconnect. Depending on workload characteristics in mobile SoCs, big.LITTLE software can schedule threads on appropriate cores and dynamically track changing performance demands. The big.LITTLE cores augmented with Neon SIMD extensions can improve performance of compute intensive game physics, graphics algorithms etc.

F. Heterogeneous multi-cores with application-specific extensions: Works such as EFFEX [8] and EVA [9] have application-specific accelerator extensions for feature extraction, tightly coupled to simple and complex cores. These works profile several feature extraction and classification algorithms to identify common kernels or computation patterns among them. Specialized processing elements to accelerate these kernels are tightly coupled to the processor. To access a rectangular region or tile of pixels as is commonly the case with feature extraction algorithms, a patch memory architecture is used. This architecture uses a software re-arrangement of a 2-d data tile into a single DRAM row. Accesses to consecutive 2-d tiles are serviced by the DRAM row buffer [8]. EFFEX offers 12X speed-up over an ARM core for HoG feature extraction algorithm.

G. Multi-cores with shared accelerators and shared memories: Works such as SARC [10], CHARM [11], AXR-CMP [12], Cogniserve [13] propose a system with multiple cores and shared accelerators. Cong et al [11,12] propose a system with general purpose cores, shared L2 cache banks and shared composable accelerator blocks, improving accelerator utilization. A core's request to use an accelerator is processed by a global controller which composes an accelerator from building blocks and allocates it to the core. Each accelerator building block island has a dedicated scratch pad memory and a DMA engine to transfer data into local scratch pads from L2, which can be overlapped with computation. The accelerator operations can be sequenced or chained by transferring data from one accelerator into another directly. Each accelerator has a TLB to work with virtual addresses. A large number of algorithms such as FAST corner detection [14], Canny edge detection, Face recognition using Local Binary Patterns [15], disparity map computation can be broken into stages with completely streaming memory access, with data flow from one stage to another enabling accelerator operations to be chained. In addition, these algorithms have a number of similar compute kernels enabling the design of accelerator building blocks.

H. Custom Accelerators: A large number of dedicated architectures have been developed for a specific application and prototyped on FPGAs. For example, Bae et. al [16] describe a platform for AIM visual saliency system

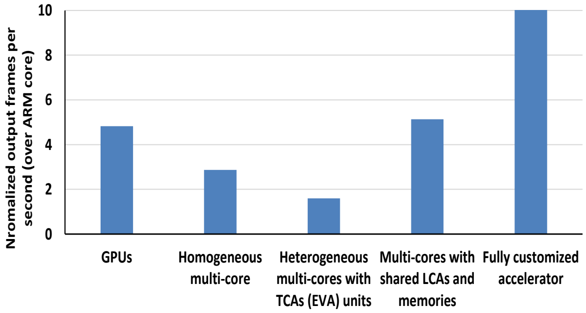


Figure 4: Ratio of output frames per second over that of an ARM core for FAST corner detection algorithm TCA = Tightly Coupled Accelerators; LCA=Loosely Coupled Accelerators

with an input camera interface prototyped on a Virtex-6 Xilinx FPGA and Park et. al [17] proposes a custom accelerator for HMAX.

Evaluation: Figure 4 shows the performance comparison for different CMOS architectures in terms of frames per second for the FAST corner detection algorithm [14]. The values are normalized over the performance of a 1GHz ARM core obtained using the GEM5 full-system simulator. Performance of FAST for homogeneous multi-core system is obtained by mapping the application on Platform-2012 [3]. The GPU used is the Nvidia GTX 280. Reported results from [9] were used for evaluating the EVA platform. We designed streaming accelerators to compute the basic primitives in FAST, such as sum of center-surround differences and non-maximal suppression. 4 out-of-order ARM CPUs were used for control flow between the accelerators. The multi-core system with shared accelerators running at 500MHz, was simulated using GEM5. The fully customized accelerator was designed with datapath and control optimized for FAST.

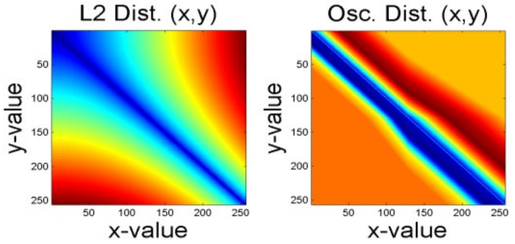


Figure 5: Comparison between L2 norm and oscillator response

III. Analog Image and Video Processing

Challenges of further scaling down the threshold voltage increasing leakage, as well as increasing uncertainty in device behavior are rendering real-time processing difficult. These challenges are exacerbated due to the rapidly increasing amount of the “Big Data” from images and videos. CMOS analog signal processing (ASP) of images becomes an attractive alternative by taking advantage of massive parallelism and unique analog circuit and architecture design [18]. Recently, a 10^{15} operations/watt analog deep machine-learning engine composed of an 8x4 array of parallel reconfigurable analog computation cells (RAC) was presented in [19], which mimics the hierarchical presentation of information in the human brain to achieve robust automated feature extraction with the accuracy comparable to the baseline software simulations. In [20], a mixed-signal VLSI array with 1.1 TMACS (10^{12} multiply-and-accumulates per second) per mW is presented, which is used in applications like pattern recognition and data compression.

In addition, several analog-based systems facilitate biological computing techniques by leveraging emerging technologies. Architectures such as IFAT [30] and NeuroDyn [31], which employ a conductance-based model of the spiking neurons found in the brain can enable non-traditional computation of problems that may be difficult or inefficiently solved using digital systems. The recent IBM SyNapse chip [45] has demonstrated unprecedented power efficiencies by using a brain-inspired neuron-based processing paradigm.

IV. Role of Emerging Technologies in Vision Systems

The increasingly significant role played by device researchers has led to the adoption of several promising device technologies in the design of new architectures for vision algorithms. Most of these new technologies exhibit physical characteristics that enable performance and power efficiency superior to existing CMOS-based architectures.

A. Magneto-metallic spin neurons: These devices have been proposed for the design of associative memory array pattern matching applications. These devices are highly energy efficient and can operate at voltages as low as 10mV [21]. These devices can operate with analog signals to determine the degree of correlation between images in terms of the magnitude of output voltage. A resistive crossbar memory designed using spin neurons is capable of highly energy efficient in-memory processing. Spin neurons can be used to realize ultra-low energy analog systems for various image processing algorithms [37]. These algorithms include edge detection, halftone compression, and digitization.

B. Patterned magnetic media: In nano-magnet logic (NML) devices, the magnetization state is used to represent binary information [22]. For example, arrangements of devices with perpendicular magnetic anisotropy could be used for image edge detection [23]. In [24], the authors report how a 2-D array of NML devices with perpendicular anisotropy can be used to implement an image filtering algorithm to remove noise from a black and white image.

C. Emerging memory technologies: There have been works that demonstrate the design of memories used for associative computing. Associative memories (AMs) are content-addressable memories that have important applications in pattern recognition, feature extraction, and classification. Several emerging memory technologies such as *Phase Change Memory* (PCM) and *Spin Torque Transfer* (STT-RAMs) [25] have been shown to be viable device options. These designs are optimized for fast lookup and hashing functionalities, which are essential in several signal and image processing applications. For every input vector, the task of the AM circuit is to find the memory vector that is the closest to the input vector using a distance metric such as the Euclidean norm. Transistor technologies such as *symmetrical-graphene-insulator-graphene FETs* (SymFETs) and *bi-layer pseudospin FETs* (BiSFETs) have I-V characteristics that are quite different from classical FETs. These characteristics make them suitable for approximating analog and multi-valued systems. As reported in [26], a SymFET-based AM has been designed and compared with an active synapse simulated in 0.13µm CMOS process. For fair comparison, a peak-to-valley ratio similar to that of the SymFET-based approach was targeted. Each CMOS-based synapse had 23 transistors (as opposed to 2 SymFETs) and consumed 76µW on average. In the SymFET design, the average power per synapse was 7.2µW.

Resistive Switching Memories, commonly known as ReRAMs, are considered to be promising candidates for future nonvolatile memory applications on account of their switching speed, scalability with technology and compatibility with existing CMOS technology. In [27], the authors demonstrate the similarity between a biological synapse and its electronic equivalent using a metal oxide ReRAM. In these devices, the resistance is varied gradually by controlling the input pulse amplitudes. In order to demonstrate adaptive learning in a neural network, these electrical synapse devices are equipped with spike-timing-dependent plasticity (STDP) functionality, a learning technique in which output depends on input data rate. More recently, Magnetic RAMs and PCMs [28] have also been adapted to similar analog architectures to enable efficient implementations of synaptic weights for the STDP exhibited by biological neurons. In [29], the authors use these learning schemes for recognition of characters in a noisy background.

D. Tunnel-FET based Accelerators: In addition to being regarded as a promising replacement to low voltage CMOS technology in general purpose processors, Tunnel FET-based customized accelerators have also been shown to demonstrate significant energy and performance benefits over conventional transistor designs. In [36], the authors have realized TFET-based accelerators used in computer vision, such as pattern matching engines. These accelerators show a 6X improvement in energy over an iso-voltage CMOS and a 30% power benefit over an iso-performance CMOS design.

E. Non-Boolean Computing: Problems such as image/pattern recognition and visual saliency can consume huge computational resources in the Boolean processing framework. This motivates the study of non-Boolean computing approaches such as locally coupled oscillators, Cellular Neural Networks (CNN) and memristor-based approximate computing.

i) Locally coupled oscillators: [32] examines the use of locally coupled oscillators in edge detection and saliency. When oscillator devices like the resonant body transistors (RBTs) [33], spin-torque nano-oscillators (STNOs) [34] and Metal Insulator Transistion (MIT) materials are coupled with each other, their outputs will finally settle down to the same phase or frequency after a settling time which depends on the difference in the input voltages to the device. Recent research on a vanadium dioxide (VO_2) MIT material, integrated with MOSFET, has shown the capability of improved image processing quality with ~20X lower power consumption over a CMOS edge/saliency detection accelerator [35]. The inherent device characteristics are observed to be similar to a distance norm of $(X^{0.5}-Y^{0.5})^2$. Figure 5 shows the comparison between an L2 distance norm, $(x^2-y^2)^{0.5}$ for a range of input values, and the response of the oscillator to the same inputs. Based on this figure, it is clear that the oscillators exhibit a comparable distance metric while the difference *(x-y)* is moderately small, with the approximation falling off more abruptly as this difference increases.

ii) Cellular Network-inspired computing: Cellular Neural Networks (CNNs) belong to a special class of Artificial Neural Networks called continuous-time Hopfield Networks. In these CNNs, all processing elements are typically connected to just nearest neighbors – which can simplify implementation. Quantitatively, for complex, 2-d image processing functions, a CNN-based processor with an area of $1.4cm^2$ and a power budget of 4.5 W could match the performance of the IBM Cellular Supercomputer with an area of 7 m^2 and a power budget of 491 KW [38].

In spite of these advantages, existing CNN implementations have several key limitations. For instance, the resolution of a state-of-the-art CNN architecture is still limited. This is because, although, an image to be processed via a CNN may typically have multiple gray levels, the output is typically binary. Recent FPGA-based approaches from Altera-Eutecus [40] are capable of processing high definition video. However, this comes at a cost of reduced functionality, increased power and reduced throughput.

Ongoing research also suggests that emerging technologies can also play an important role in improving the power/efficiency of CNNs. Non-linear devices such as resonant tunneling diodes (RTDs) [41], and more recently TFETs [42] have been introduced into CNN circuitry to solve binary classification problems by eliminating the required output transport function hardware.

Additionally, designs for TFET-based CNNs have also been proposed that could be used to solve more complex, multi-valued classification problems. When studying a slippage detection problem, where tactile data is treated as an image, a conventional binary CNN requires 5 processing steps (i.e., template operations) and 2 hardware data paths [43]. Alternatively, a TFET-based circuit with ternary outputs can solve the same problem with just 3 computational steps and 50% less hardware. Further, a ternary CNN cell is expected to dissipate 70X less energy for the detection task.

In CNNs, most template operations leverage linear relationships between cells. However, non-linear templates often reduce the number of programming steps required to solve a particular problem, when compared to an algorithm that employs only linear templates. Initial work [44] suggests that non-linearities associated with SymFETs can be used to efficiently realize non-linear templates. As an example, to perform thresholding on an interval using a band pass filter, three sequential linear operations are needed. The same task can be accomplished with a single template operation when SymFETs are employed. Thus, there are two potential sources of simultaneous improvement – a reduction in template operations and hardware complexity required to realize non-linear operations.

iii) Memristor-based Approximate Computing: As processing power budgets continue to tighten, non-traditional techniques such as approximate computing are becoming more popular. In this paradigm, rather than trade chip area or performance, architectures leverage the accuracy of computation for saving power. Memristor-based computing architectures exploit the non-determinism of resistive memories to produce efficient, high performance systems, which yield approximate, rather than definitive results. These systems are shown to be capable of highly power-efficient computation, consuming up to 300X less power and over 400X improved performance compared to general purpose CPUs [39].

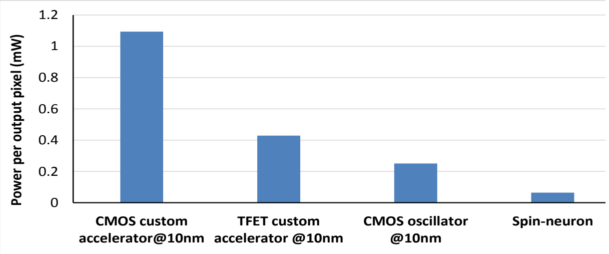


Figure 6: Comparison of power per output pixel across different technology-based systems for a pattern matching accelerator

Evaluation: Figure 6 shows the power per output pixel, for various technology-based accelerators described in this section when implementing a pattern matching algorithm. A distance compute accelerator was designed and synthesized using Synopsys tools with the Synopsys SAED 32nm libraries. We obtained power from Synopsys design compiler and scaled it down to 10nm technology using factors reported in [1] and projections from ITRS (International Technology Roadmap for Semiconductors). We synthesized the TFET accelerator using the 22nm library described in [36] and used TCAD simulations to obtain corresponding power numbers at the 10nm node. Results for the spin neuron-based implementation were reported in [21].

V. CONCLUSION

This paper provides a comprehensive survey of the various techniques used to design image processing systems. From this work, it is evident that the potential for improvements in performance and energy efficiency lies at the convergence of advances in device technology, analog and digital circuit design and system architectures and algorithms.

VI. ACKNOWLEDGMENTS

This work was supported by the Center for Low Energy Systems Technology (LEAST), an SRC program sponsored by MARCO and DARPA and by the NSF award 1317560.

REFERENCES

[1] Esmaeilzadeh, H et al, “Dark silicon and the end of multicore scaling”. In ISCA, 2011.

[2] www.tilera.com/sites/default/files/productbriefs/TILE-Gx8072_PB041-03_WEB.pdf

[3] Melpignano, D., et al. "Platform 2012: a many-core computing accelerator for embedded SoCs" DAC 2012.

[4] Park, H., et al, “Polymorphic pipeline array: a flexible multicore accelerator with virtualized execution for mobile multimedia applications”. In MICRO 2011.

[5] http://infocenter.arm.com/help/topic/com.arm.doc.dht0002a/DHT0002A_introducing_neon.pdf

[6] Lee Y, et al “Exploring tradeoffs between programmability and efficiency in data-parallel accelerators”, ACM SIGARCH Comp. Arch News

[7] “Improving Energy Efficiency and Performance in Mobile Devices”, Brian J., Nov 2013.

[8] Clemons, J, et al. "EFFEX: an embedded processor for computer vision based feature extraction." Design Automation Conference (DAC), 2011.

[9] Clemons, J et al, “EVA: An efficient vision architecture for mobile systems”. In CASES 2013

[10] Ramirez, Alex, et al. "The SARC architecture." IEEE Micro 2010.

[11] Cong, J et al, CHARM: a composable heterogeneous accelerator-rich microprocessor”. In ISLPED 2012

[12] Cong, J et al, “AXR-CMP: Architecture support in accelerator-rich CMPs”. SoC Architecture, Accelerators and Workloads (SAW), 2011.

[13] Iyer, R., et al “Cogniserve: Heterogeneous server architecture for large-scale recognition”. Micro, IEEE, 31(3), 20-31.

[14] Rosten, E., & Drummond, T. (2006). “Machine learning for high-speed corner detection”. Computer Vision–ECCV 2006, 430-443.

[15] Ahonen, T., Hadid, A., and Pietikainen, M. Face Recognition with Local Binary Patterns. Computer Vision - ECCV 2004 (2004), 469–481

[16] Bae, S., et al. "An FPGA implementation of information theoretic visual-saliency system and its optimization." FCCM 2011.

[17] Park, Sungho, et al. "System-on-chip for biologically inspired vision applications." IPSJ Transactions on System LSI Design Methodology, 2012

[18] Yu Chi, et al “Intensity Histogram CMOS Image Sensor for Adaptive Optics,” ISCAS 2010

[19] Junjie et al, “A 1TOPS/W Analog Deep Machine-Learning Engine with Floating-Gate Storage in 0.13µm CMOS,” ISSCC 2014

[20] R Karakiewicz, et al “1.1 TMACS/mW Fine-Grained Stochastic Resonant Charge-Recycling Array Processor,” IEEE Sensors Journal, 2012.

[21] Mrigank S et al, "Ultra low power associative computing with spin neurons and resistive crossbar memory”, in DAC '13.

[22] M. T. Niemier et al., "Nanomagnet Logic: Progress Toward System-Level Integration," J. Phys. Con. Mat., vol. 23, p. 493202, 2011.

[23] M. Niemier, et al., "Boolean and Non-Boolean Architectures for Out-of-Plane Nanomagnet Logic," Procedings of the International Workshop on Cellular Nanoscale Networks and their Applications, pp. 1-6, 2012

[24] A. Popovici and D. Popovici, "Cellular Automata in Image Processing," in Int. Symp. on Mathematical Theory of Networks and Systems, 2002.

[25] Q. Guo et al. A resistive TCAM accelerator for data-intensive computing. MICRO 2013.

[26] B. Sedighi, et al, "Nontraditional Computation using Beyond-CMOS Tunneling Devices," under review in JETCAS, 2014.

[27] D. Kuzum, et al, “Low-Energy Robust Neuromorphic Computation Using Synaptic Devices,” IEEE Trans. Electron Devices, Dec 2012.

[28] Jackson, B., et al, "Nanoscale electronic synapses using phase change devices." Journal on Emerging Technologies in Computing Systems 2013.

[29] V Narayanan, et al, "Video Analytics Using Beyond CMOS Devices", Design Automation & Test in Europe (DATE), 2014

[30] Vogelstein, R. et al. "Dynamically reconfigurable silicon array of spiking neurons with conductance-based synapses." Neural Networks, IEEE Transactions on 18.1 (2007): 253-265.

[31] Yu, Theodore, and Gert Cauwenberghs. "Analog VLSI biophysical neurons and synapses with programmable membrane channel kinetics." Biomedical Circuits and Systems, IEEE Transactions on, 2010.

[32] M. Cotter et al, “Computational architectures based on coupled oscillators,” NewCAS 2014;

[33] D. Weinstein and S. A. Bhave, "The Resonant Body Transistor," Nano Letters, pp. 1234-1237, 2010.

[34] J. Akerman, "Spin torque oscillators," in International Conference on Advanced Materials, 2009.

[35] N. Shukla, et al, “Pairwise coupled hybrid Vanadium dioxide-MOSFET (HVFET) Oscillators for non-Boolean associative computing,” IEDM 2014

[36] Swaminathan, K, et al. "Modeling steep slope devices: From circuits to architectures”, DATE, 2014.

[37] M . Sharad et. al., “Ultra Low Energy Analog Image Processing Using Spin Based Neurons”, Nanoarch 2012.

[38] Roska, T., "Cellular wave computers for brain-like spatial-temporal sensory computing," *Circuits and Systems Magazine, IEEE*, 2005.

[39 Li, B. et. al., "Memristor-based Approximate Computation", ISLPED 2013, pp-242-247

[40] Eutecus. (2012). *Multi-core Video Analytics Engine (MVE™).*

[41] P. Mazumder, S. Li, and I. Ebong, "Tunneling-Based Cellular Nonlinear Network Architectures for Image Processing," VLSI Systems, Trans 2009.

[42] I. Palit, et al, "TFET based Cellular Neural Network (CNN) Architectures," in ISLPED 2013.

[43] A. Kis, et al, "3D tactile sensor array processed by CNN-UM: a fast method for detecting and identifying slippage and twisting motion: Research Articles," Int. J. Circuit Theory Appl, 2006.

[44] A. Horváth, et al., "Architectural Impact of Emerging Transistors," IEEE NEWCAS, 2014.

[45] T. Simonite, “IBM Chip Processes Data Similar to the Way Your Brain Does”, MIT Technology Review, Aug, 2014