Intelligent Vision Systems: Exploring the State-of-the-Art and Opportunities for the Future

Siddharth Advani*, Srinidhi Kestur*,‡, Vijaykrishnan Narayanan*
*The Pennsylvania State University, USA
‡Broadcom Corporation

***Abstract*— Vision and Video applications are becoming ubiquitous in mobile and embedded systems. The advent of wearable devices which require capabilities for real-time video analytics and prolonged battery lifetimes is further driving the need for innovative system designs with low-power, reliability and high performance. Further, the increasing resolution of image sensors in these mobile systems places an increasing demand on both the memory storage as well as the computational power. Such stringent requirements have given rise to accelerator-rich architectures in system-on-chips, where the primary computational burden is handled by dedicated hardware accelerators.**

In this paper we explore existing Vision accelerators and analyze their architecture, performance and scalability for different datasets and applications. The applications evaluated in this work are neuro-biologically inspired algorithms for object detection, object recognition and activity recognition which are complex, compute-intensive and bandwidth-intensive. This paper further analyzes the reliability of such embedded vision systems in terms of robustness of performance and energy efficiency under different application scenarios. Specifically, this work discusses the opportunities to improve energy efficiency by minimizing DRAM refreshes and explores techniques to exploit algorithmic resilience to minimize power consumption while maintaining reliable system accuracy and performance.

I. Introduction

The workings of the brain have intrigued researchers across various spectrums of science - from neuroscience to computer science. While the cortex still remains an enigma to the community, the visual cortex is a more finely understood system and many mathematical models mimicing the human visual system (HVS) have been proposed. Some of the early work in vision focused on understanding how primal capabilities of vision trigger higher modalities such as object recognition [1]. Progressively, object recogntion models based on the simple and complex cells in the cortex were developed [2]. To understand task-driven vision, attention models using salient features of a visual scene were proposed [3], [4]. More recent work focuses on understanding the impact of attention under the influence of multiple cues [5].

Most of these vision models are computationally intensive that require frequent accesses to memory due to large matrix operations that run either in a feed-forward or an iterative manner. Running these workloads in real-time is a necessary constraint that needs to be met by the underlying system, but is becoming a daunting challenge with increasing resolutions of display panels coupled with improved camera sensors.

Given the challenges and opportunities in neuromorphic computing, many have embarked upon developing systems for smart vision applications. Synopys recently launced EV544 - a Convolutional Neural Network (CNN) based processor [6]. The SpiNNaker project has evolved from being a massively parallel representation of the human brain to now being used as a tool to further advance studies in neuroscience and robotics [7]. In a similar league, the True North chip [8] meanders away from the traditional von Neumann architecture and uses 4096 parallel and distributed cores is an event-driven framework for solving problems in vision and audition.

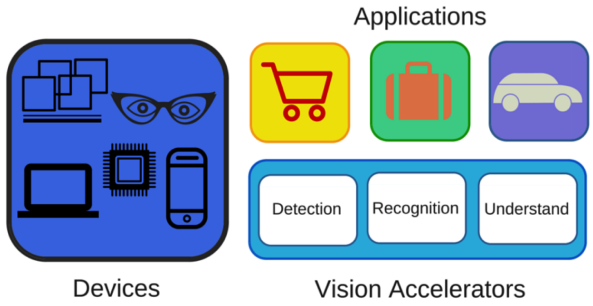


Fig. 1. Interaction between a vision pipeline and the potential platforms to which it can be mapped depending on the application demands.

Fig. 1 illstrates the interaction between compute devices and vision accelerators when targeting various applications. A common vision pipeline involves parsing the visual scene and extracting objects or regions of interest (RoIs). This is carried out in the object detection stage. Once regions are extracted they are sent to a recognition stage to identify what the object is. Having figured out whether the object is of interest, further options can be explored. For example, if the object is a person, activity or pose estimation can be triggered. The application workload usually will decide the choice of the compute device. For example, if a user is in a retail store and would like to use a smart visual-assist device, a wearable small form-factor device would be ideal. However, if this is an automotive-assist system, a larger device may be engaged. If a security application is being deployed at an airport, then a large server-scale architecture would be needed to handle the sheer volume of data being generated every minute.

The main contributions of this paper are:

- To usher in the next wave of technology, we explore the current state-of-the-art in embedded vision accelerators and lay emphasis on key insights when designing such accelerators.
- With scaling technology paving the way for approximate computing, we exploit an increasingly powerful

property of most vision algorithms - reliability to noise. We show that for an object recognition system, we can save upto X% refresh power and reduce the number of multipliers by 7× in a single module when using DRAMs for memory storage while maintaining a 1% error bound on accuracy.

The rest of this paper is organized as follows: In Section II, we provide an overview of vision-based architectures and the corresponding state-of-the-art. Section III describes a robust object recogntion pipeline. Finally, we conclude with Section IV.

II. Vision Accelerators

Due to the capacity of human vision systems for highly complex processing at very low power, many brain-inspired algorithms and architectures have been proposed to emulate the human visual cortex. [9], [10], [11].

Digital signal processors (DSPs) have been a universally accepted alternative to general purpose CPUs for seamless multimedia processing. The Qualcomm Hexagon DSP instruction set architecture (ISA) contains numerous special-purpose instructions designed to accelerate key multimedia kernels such as sliding window filters [12]. Heterogeniety has often been considered as a viable solution to handle the increasingly varying nature of workloads that need to be run today [13]. Texas Instruments has a heterogenous multi-core DSP targeted for real-time vision applications based on their Keystone architecture. To tackle the diverse field of vision many other heterogenous architectures have been proposed that take advantage of customized flows for regular systolic operations while using the traditional von Neumann architecture for handling control logic and other irregular data operations. A heterogenous server architecture consisting of many small cores for low power and high throughput coupled with custom hardware accelerators was designed in [14]. In [15], the authors explored architectural heterogeniety by using customized data-flows for many vision-based applications targeted at retail, security, etc.

Most vision applications demand high throughput and specialized accelerators have shown to be extremely performance-friendly for computationaly intensive tasks such as face detecion [16], pedestrian detection [17], object recognition [18] and object detection [19]. In [20], the authors propose a benchmarking suite - VISBench (Visual, Interactive, Simulation Benchmarks) - and find that a MIMD rather than a SIMD architecture gives better performance.

Even though Convolutional Neural Networks (CNNs) were explored in the early 1990s for vision applications [21], they have resurfaced again after a long hiatus and become extremely popular in the past couple of years. This successful comeback can be attributed to two major phenomena: (1) the existence of large amount of data (needed to train the network well) with the evolution of the digital era, and (2) the development of custom hardware (required for acceleration) now being used for CNNs.

In the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) conducted in 2012, the winning team trained a CNN consisting of five convolutional and three fully-connected layers. Importantly, the depth of the CNN is critical to its recognition capabilities since the authors found that removing any convolutional layer resulted in inferior performance [22]. This CNN would need more than 80 million operations and over 100,000 data transfers [23].

More recent and advanced CNN architectures have 10 to 20 layers of Rectified Linear Units, hundreds of millions of weights, and billions of connections between units. The reader is pointed to [24] for insights on deep architectures in general and [25] for CNN-based learning and their recent advances.

From a systems perspective, [26] mapped an earlier Convolutional Network based face-detection task onto custom hardware. More recently, [10] recently proposed an architecture for CNNs and Deep Neural Networks (DNNs) that minimized memory transfers thus achieving high throughput with small area, power and energy footprint. [27] furthered this by proposing a training and inference accelerator capable of providing GPU-like bandwidth in ASIC-like power budgets.

III. Reliability

Reliability is being explored at different layers of abstraction; from devices [28], [29], [30] to memory [31] to algorithms. At a circuit-level, [32] uses a conditional probability approach for modeling reliability in combinational circuits.

A. Introduction

In this section, we evaluate the capabilities of a popular visual object recognition algorithm - HMAX - and exploit the potential to save power and reduce computational load.

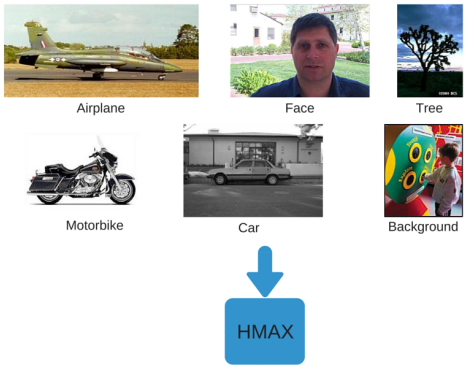
B. HMAX

HMAX is a hierarchical visual object recognition model that has been used in various embedded real-time applications [11], [18].

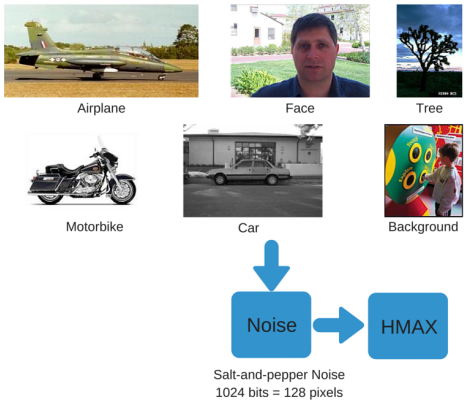
C. Exploiting Resiliency for Power Benefits

So far we have surveyed the landscape of vision systems that enhance the performance and energy efficiency of the computational fabrics. However, memory is an integral part of most systems today and contributes between 10-30% of the overall power of embedded video systems and mobile phones [33]. The increasing memory size in new generations of embedded systems and the use of stacked 3D architectures that increase on-chip temperatures have made researchers look at saving on memory refresh energy. New power-efficient techniques such as Low Power Auto Self Refresh, Temperature Controlled Refresh, Refresh Pausing, Fine Granularity Refresh and Data Bus Inversion have been introduced in new memory standards such as DDR4 [34].

Tuning DRAM refresh based on the data characteristics has been proposed as early as 1998 [35]. Many recent works have looked at tackling the increasing refresh power in other different ways [36], [37]. In [38], the authors looked at reducing refresh power on multimedia workloads. Recently, in [39], the authors showed that in real-time embedded vision applications, refresh power can dynamically be changed based on autonomously tagging data with logical labels.



(a) Baseline



(b) Noise

Fig. 2. HMAX resilience to errors. Six classes from CalTech101 were used.

In this section, we explore the resiliency of HMAX to bit errors that can then be used to choose the refresh rate for DRAMs when these images are stored. Fig. 3 illustrates the classification accuracy of HMAX as a function of the pixel errors introduced in each image.

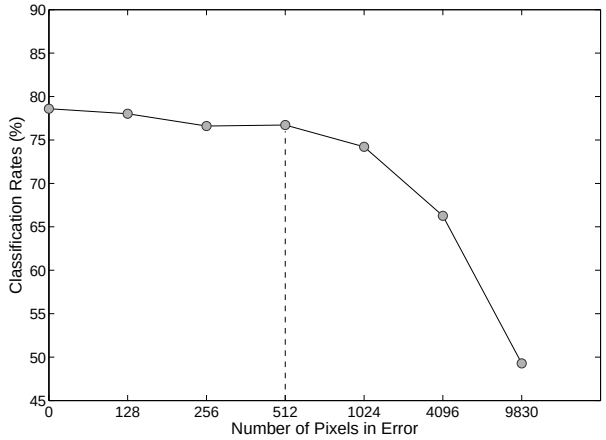


Fig. 3. HMAX resilience to errors. Six classes from CalTech101 were used.

D. Exploiting Reseliency for Compute Benefits

Image reconstruction is an important processing technique in image processing and computer vision applications. Most object recognition algorithms use a multi-scale pyramid to make it scale invariant. For example, HMAX uses an image pyramid having 11 scales (including base scale) with a scale factor of $2^{1/4}$ and uses a bicubic interpolation technique to generate the image pyramid. The input image is passed through this image pyramid before computing the "S1" layer of HMAX.

Many architectures have been proposed to support linear and non-linear interpolation techniques [40].

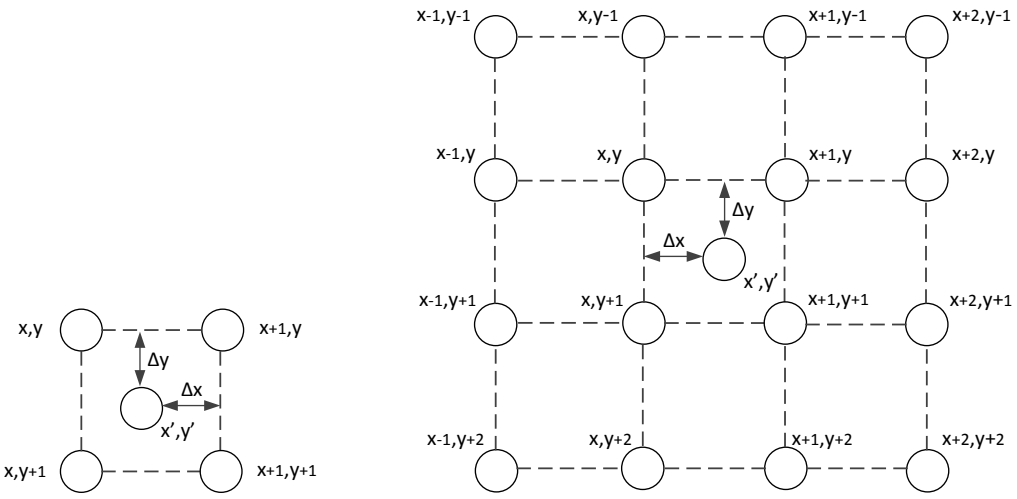
Given a pixel $I(x, y)$, an interpolated pixel $I'(x', y')$ using bilinear interpolation is given by (1). By definition, this involves computing two linear interpolation in x and y directions and requires eight multiplications. Figure 4(a) illustrates an interpolated pixel using four neighboring pixels.

$$\begin{aligned} I'(x', y') = & I(x, y) \times (1 - \Delta x) \times (1 - \Delta y) + \\ & I(x + 1, y) \times \Delta x \times (1 - \Delta y) + \\ & I(x, y + 1) \times (1 - \Delta x) \times \Delta y + \\ & I(x + 1, y + 1) \times \Delta x \times \Delta y \end{aligned} \tag{1}$$

Using bicubic interpolation, the same interpolated pixel $I'(x', y')$ is given by (2) where R_c denotes a bicubic interpolation function. The computation requires 56 multiplications in all and Figure 4(b) shows the interpolated pixel using sixteen neighboring pixels.

$$I'(x', y') = \sum_{m=-1}^{2} \sum_{n=-1}^{2} I(x + m, y + n) R_c(m - \Delta x) R_c(-(n - \Delta y)) \tag{2}$$

In this section we explore the potential savings in computational work needed to be done while not compromising on accuracy. In the embedded version, compute resources are very costly. Saving a few resources can result in being able to fit a design in a particular form-factor or may cause the design to overflow into the next larger generation of devices. We explored the capablity of HMAX to correctly recognize objects using bilinear interpolation in the image pyramid. We used all 101 classes of CalTech101 for this purpose. It should be noted that using the original bicubic interpolation technique, we achieve 54% accuracy on the said dataset. This is in confirmation with the results shown in [2]. We then ran the experiment using bilinear interpolation and found the impact of this is a 1% loss in accuracy. Also, instead of 56 multipliers (bicubic interpolation), we would need just eight multipliers (bilinear interpolation). Table I shows the results and the savings have a significant impact on area and performance of the accelerated system.



(a) Bilinear Interpolation

(b) Bicubic Interpolation

Fig. 4. Interpolation techniques. In (a), four while in (b), 16 neighboring pixels are used for interpolation.

TABLE I. IMPACT OF INTERPOLATION TECHNIQUES

System	Algorithm	Accuracy	Multipliers
HMAX	Bicubic	54%	56
HMAX	Bilinear	53%	8

IV. CONCLUSION

In this paper we extensively survey a plethora of vision-based systems targeted for real-time applications. With shrinking technology, reliability will become exceedingly challenging and approximate computing will have more of a role to play when designing compute systems. We exploit the inherent robustness in vision algorithms and show that it can help to reduce both power and compute resources, both of which are valuable when designing such systems.

ACKNOWLEDGEMENTS

This work is supported in part by NSF Expeditions: Visual Cortex on Silicon CCF 1317560. The work is also supported through infrastructure provided by NSF Award 1205618.

REFERENCES

[1] D. Marr, "Early Processing of Visual Information," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 275, no. 942, pp. 483–519, 1976.

[2] J. Mutch and D. G. Lowe, "Object class recognition and localization using sparse features with limited receptive fields," *IJCV*, 2008.

[3] N. D. B. Bruce and J. K. Tsotsos, "Saliency Based on Information Maximization," *Advances in Neural Information Processing Systems*, vol. 18, pp. 155–162, 2006.

[4] L. Itti and C. Koch, "Computational Modelling of Visual Attention," *Nature Reviews Neuroscience*, 2001.

[5] M. Bay and B. Wyble, "The Benefit of Attention is not Diminished when Distributed Over Two Simultaneous Cues," *Attention, Perception, and Psychophysics*, vol. 76, no. 5, pp. 1287–1297, 2014.

[6] J. Campbell and V. Kazantsev, "Using an Embedded Vision Processor to Build an Efficient Object Recognition System," in *DesignWare IP White Papers*, May 2015.

[7] S. Furber, F. Galluppi, S. Temple, and L. Plana, "The SpiNNaker Project," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652–665, May 2014.

[8] A. Cassidy *et al.*, "Real-time scalable cortical computing at 46 giga-synaptic ops/watt with 100x speedup in time-to-solution and 100,000x reduction in energy-to-solution," in *High Performance Computing, Networking, Storage and Analysis, SC14: International Conference for*, Nov 2014, pp. 27–38.

[9] A. Nere, A. Hashmi, and M. Lipasti, "Profiling Heterogeneous Multi-GPU Systems to Accelerate Cortically Inspired Learning Algorithms," in *IPDPS*, 2011.

[10] T. Chen, J. Wang, Y. Chen, and O. Temam, "DianNao : A Small-Footprint High-Throughput Accelerator for Ubiquitous Machine-Learning," in *ASPLOS*, 2014.

[11] S. Kestur *et al.*, "Emulating Mammalian Vision on Reconfigurable Hardware," in *IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, April 2012.

[12] L. Codrescu *et al.*, "Hexagon dsp: An architecture optimized for mobile multimedia and communications," *Micro, IEEE*, vol. 34, no. 2, pp. 34–43, Mar 2014.

[13] E. Chung, P. Milder, J. Hoe, and K. Mai, "Single-Chip Heterogeneous Computing: Does the Future Include Custom Logic, FPGAs, and GPGPUs?" in *Microarchitecture (MICRO), 2010 43rd Annual IEEE/ACM International Symposium on*, Dec 2010, pp. 225–236.

[14] R. Iyer *et al.*, "CogniServe: Heterogeneous Server Architecture for Large-Scale Recognition," *Micro, IEEE*, May 2011.

[15] N. Chandramoorthy *et al.*, "Exploring Architectural Heterogeneity in Intelligent Vision Systems," in *International Symposium on High Performance Computer Architecture (HPCA)*, Feb 2015.

[16] D. Hefenbrock, J. Oberg, N. Thanh, R. Kastner, and S. Baden, "Accelerating Viola-Jones Face Detection to FPGA-Level Using GPUs," in *Field-Programmable Custom Computing Machines (FCCM), 2010 18th IEEE Annual International Symposium on*, May 2010, pp. 11–18.

[17] A. Suleiman and V. Sze, "Energy-efficient HOG-based Object Detection at 1080HD 60 fps with Multi-Scale Support," in *IEEE Workshop on Signal Processing Systems*, Oct 2014.

[18] A. Maashri *et al.*, "Accelerating neuromorphic vision algorithms for recognition," in *DAC*. ACM Press, 2012, p. 579.

[19] S. Bae *et al.*, "An FPGA Implementation of Information Theoretic Visual-Saliency System and Its Optimization," in *IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, May 2011.

[20] A. Mahesri, D. Johnson, N. Crago, and S. Patel, "Tradeoffs in Designing Accelerator Architectures for Visual Computing," in *Microarchitecture, 2008. MICRO-41. 2008 41st IEEE/ACM International Symposium on*, Nov 2008, pp. 164–175.

[21] S. Lawrence, C. Giles, and A. C. Tsoi, "Convolutional Neural Networks for Face Recognition," in *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, Jun 1996, pp. 217–222.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105.

[23] XCell, "Machine Learning in the Cloud: Deep Neural Networks on FPGAs," Available: http://issuu.com/xcelljournal/docs/xcell_journal_issue_92/46?e.

[24] Y. Bengio, "Learning Deep Architectures for AI." Now Publishers, 2009.

[25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," in *Nature*, May 2015, pp. 436–444.

[26] C. Farabet, C. Poulet, J. Han, and Y. LeCun, "CNP: An FPGA-based processor for Convolutional Networks," in *International Conference on Field Programmable Logic and Applications (FPL)*, Aug 2009.

[27] Y. Chen *et al.*, "DaDianNao: A Machine-Learning Supercomputer," in *MICRO*, Dec 2014, pp. 609–622.

[28] S. Datta, H. Liu, and V. Narayanan, "Tunnel FET technology: A reliability perspective," *Microelectronics Reliability*, vol. 54, no. 5, pp. 861 – 874, 2014.

[29] N. Agrawal, H. Liu, R. Arghavani, V. Narayanan, and S. Datta, "Impact of Variation in Nanoscale Silicon and Non-Silicon FinFETs and Tunnel FETs on Device and SRAM Performance," *Electron Devices, IEEE Transactions on*, vol. 62, no. 6, pp. 1691–1697, June 2015.

[30] R. Pandey *et al.*, "Tunnel Junction Abruptness, Source Random Dopant Fluctuation and PBTI Induced Variability Analysis of $GaAs_{0.4}Sb_{0.6}/In_{0.65}Ga_{0.35}As$ Heterojunction Tunnel FETs," *IEEE International Electron Devices Meeting*, (Accepted) 2015.

[31] L. Chen and Z. Zhang, "MemGuard: A low cost and energy efficient design to support and enhance memory system reliability," in *Computer Architecture (ISCA), 2014 ACM/IEEE 41st International Symposium on*, June 2014, pp. 49–60.

[32] C. Chen and R. Xiao, "A Fast Model for Analysis and Improvement of Gate-Level Circuit Reliability," *Integration, the VLSI Journal*, 2015.

[33] A. Carroll and G. Heiser, "An Analysis of Power Consumption in a Smartphone," in *Usenix Annual Technical Conference*, 2010.

[34] "JEDEC DDR3 and DDR4 SDRAM Standard," 2012.

[35] T. Ohsawa, K. Kai, and K. Murakami, "Optimizing the dram refresh count for merged DRAM/logic LSIs," in *ISLPED*, 1998.

[36] J. Liu, B. Jaiyen, R. Veras, and O. Mutlu, "RAIDR: Retention-Aware Intelligent DRAM Refresh," in *ISCA*, 2012.

[37] J. Stuecheli, D. Kaseridis, H. Hunter, and L. John, "Elastic Refresh: Techniques to Mitigate Refresh Penalties in High Density Memory," in *MICRO*, 2010.

[38] S. Liu, Pattabiraman K., T. Moscibroda, and B. Zorn, "Flikker: Saving DRAM Refresh-power through Critical Data Partitioning," in *ASLPOS*, 2011.

[39] S. Advani *et al.*, "Refresh Enabled Video Analytics (REVA): Implications on Power and Performance of DRAM Supported Embedded Visual Systems," in *Computer Design (ICCD), 2014 32nd IEEE International Conference on*, Oct 2014, pp. 501–504.

[40] S. Kestur, K. Irick, S. Park, A. Al Maashri, V. Narayanan, and C. Chakrabarti, "An Algorithm-Architecture Co-Design Framework for Gridding Reconstruction using FPGAs," in *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, June 2011, pp. 585–590.