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

Many chip-makers are now earmarking a significant amount of research effort for vision-based processors. Texas Instruments offers a heterogenous multi-core DSP for real-time vision applications using their Keystone architecture. Recently Freescale Semiconductor unveiled a vision system-on-chip - S32V - for accident-free-cars. Camera-friendly wearable devices like Google Glass are demanding better power efficiencies, improved performance and more powerful capabilities from the underlying technologies. Synopys recently launced EV544 - a Convolutional Neural Network (CNN) based processor [1].

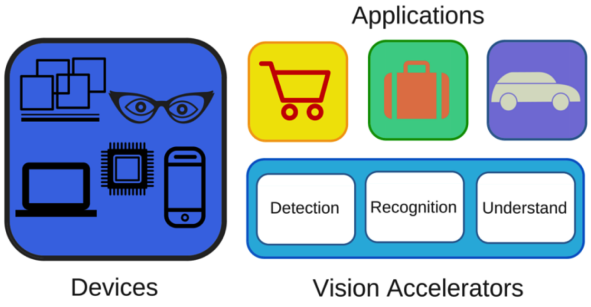

Fig. 1.

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. It should be noted that in all these applications, real-time is a necessary constraint that needs to be met by the underlying system.

In the context of real-time vision applications, single-class object detection is a highly computationally intensive task. To robustly detect an object in an image that may appear at arbitrary position and scale involves (1) extracting optimized features that aptly describe the object and (2) searching the image in a sliding window fashion for the presence of particular configurations of the features that are indicative of the object's presence. This exhaustive search is compounded by objects that exhibit high appearance variability in shape, color and size. But, for visual-assist systems, the ability to perform such a task is imperative. For example, in a visual driving assist system, an approaching vehicle or a passing pedestrian needs to be detected with minimal latency, minimum false positives, and maximum accuracy. On the other hand, a wearable visual prosthesis device needs to augment the visual cognition of the user in diverse and vastly unconstrained environments for extended periods of time.

In this paper, we focus on xyz. To augment the next generation of wearables, we lay emphasis on abc. The main contributions of this paper are:

- We survey the state-of-the-art.
- We exploit reliability.

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. Related Work

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. [2], [3], [4].

In [5], the authors explored architectural heterogeniety by using customized data-flows for many vision-based applications targeted at retail, security, etc.

Even though Convolutional Neural Networks (CNNs) were explored in the early 1990s for vision applicationsi [6], 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 [7]. This CNN would need more than 80 million operations and over 100,000 data transfers [8].

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 [9] for insights on deep architectures in general and [10] for CNN-based learning and their recent advances.

From a systems perspective, [11] mapped an earlier Convolutional Network based face-detection task onto custom hardware. More recently, [3] 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.

Most works in this domain have focused mainly on enhancing the performance and energy efficiency of the computational fabrics and do not address the inefficiencies of the main memory system. The memory system contributes between 10-30% of the overall power of embedded video systems and mobile phones [12]. The increasing memory size in new generations of embedded systems and the use of stacked 3D architectures that increase on-chip temperatures have drawn increasing attention on reducing the memory refresh energy. Consequently, there have been sustained efforts to introduce new power-efficient techniques such as Low Power Auto Self Refresh, Temperature Controlled Refresh, Refresh Pausing, Fine Granularity Refresh and Data Bus Inversion in new memory standards such as DDR4 [13]. Tuning DRAM refresh based on the data characteristics has been proposed as early as 1998 [14]. Recently, a software approach, termed as *Flikker* was proposed that relies on the user to annotate critical and non-critical parts [15]. It also allows refresh rates to be different for critical and non-critical sections of the memory and conserves the refresh energy.

III. Reliability

Reliability is being explored at different layers of abstraction; from devices [16], [17], [18] to memory [19] to algorithms. At a circuit-level, [20] uses a conditional probability approach for modeling reliability in combinational circuits.

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.

A. Power

Many previous works have looked at tackling the increasing refresh power in DRAMs [21], [22]. In [15], the authors looked at reducing refresh power on multimedia workloads. Recently, in [23], the authors showed that in real-time embedded vision applications, refresh power can dynamically be changed based on autonomously tagging data with logical labels.

HMAX is a hierarchical visual object recognition model that has been used in various embedded real-time applications [4], [24]. In this work, 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. 2 illustrates the classification accuracy of HMAX as a function of the pixel errors introduced in each image.

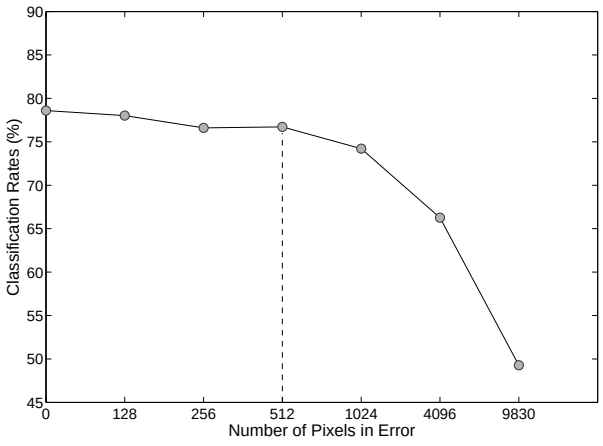

Fig. 2. HMAX resilience to errors. Six classes from CalTech101 were used.

B. Compute

In this section we explore the potential savings in computational work needed to be done while not compromising on accuracy. The input image is passed through an image pyramid before computing the "S1" layer of HMAX. The original HMAX model uses a bicubic interpolation technique to generate the image pyramid. 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 [25]. We then ran the experiment using bilinear interpolation and found that the accuracy drops by just 1%. The impact of this is that instead of 44 multipliers (bicubic interpolation), we would need just eight multipliers (bilinear interpolation). Table I shows the results.

TABLE I. INTERPOLATION TECNIQUES

System	Algorithm	Accuracy	Resources
HMAX	Bicubic	54%	44
HMAX	Bilinear	53%	8

IV. CONCLUSION

In this work, we showcase alpha,betta,gamma

Future work entails uvw.

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

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

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

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

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

[6] 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.

[7] 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.

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

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

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

[11] 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.

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

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

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

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

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

[17] 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.

[18] 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.

[19] 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.

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

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

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

[23] 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.

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

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