High-Throughput Fixed-Point Object Detection on FPGAs

Xiaoyin Ma *, Walid Najjar † and Amit Roy-Chowdhury *
*Department of Electrical Engineering
†Department of Computer Science & Engineering
University of California, Riverside
Riverside, CA 92521, USA

***Abstract*—** Computer vision applications make extensive use of floating-point number representation, both single and double precision. The major advantage of floating-point representation is the very large range of values that can be represented with a limited number of bits. Most CPU, and all GPU, designs have been extensively optimized for short latency and high-throughput processing of floating-point operations. On an FPGA, the bit-width of operands is a major determinant of its resource utilization, the achievable clock frequency and hence its throughput. By using a fixed-point representation with fewer bits, an application developer could implement more processing units and a higher-clock frequency and a dramatically larger throughput. However, smaller bit-widths may lead to inaccurate or incorrect results.

Object and human detection are fundamental problems in computer vision and a very active research area. In these applications a high throughput and an economy of resources are highly desirable features allowing the applications to be embedded in mobile or field-deployable equipment. The Histogram of Oriented Gradients (HOG) algorithm [1], developed for human detection and expanded to object detection, is one of the most successful and popular algorithm in its class. In this algorithm, object descriptors are extracted from detection window with grids of overlapping blocks. Each block is divided into cells in which histograms of intensity gradients are collected as HOG features. Vectors of histograms are normalized and passed to a Support Vector Machine (SVM) classifier to recognize a person or an object.

HOG algorithm for object detection achieves a high detection accuracy but delivers just under 1 frame-per-second (fps) on a high-end CPU. Previous FPGA implementations [2]–[4] of HOG either do not use fixed-point, or used fixed-point with large word size that results in large FPGA resource usage. In this work we explore the effects of reduced bit-width on the accuracy and performance of the HOG algorithm when implemented on an FPGA. We have used six pedestrian detection benchmarks suites, totaling 10,000 frames, all with known ground-truth to evaluate our fixed-point HOG detection [5]–[8].

In traditional fixed-point object detection implementation, the ideal fixed-point data size is determined by applying both fixed-point and floating-point object detectors to windows known as objects or background windows for detection rate comparison. However, this approach may not correctly predict the actual detection performance when consider the entire frame across multiple image scales. Usually a post-processing step is performed on all positive windows across the image at all scales to merge nearby windows. This step can significantly reduce the number of false positive windows found by the detector. Thus, our work applied the full image evaluation methodology proposed by Dollar *et al.* [7], [9] for the fixed-point detection evaluation used in this paper. By applying the state of the art computer vision object detection evaluation metric on 10,000 frames, we show that reducing the bit-width to 12 bits preserves the same detection accuracy as the original floating-point and even enhances it. Finally, we chose 13-bits in our hardware implementation as it provides a balance between precision and recall and consistent performance across all benchmarks.

The FPGA implementation of HOG on the Convey Computers HC-2ex using a single Xilinx Virtex-6 LX760 using 13-bits fixed-point is 271 stage pipeline occupying 3.3% of the FPGA resources. It shows a 4.7x and 11.6x reductions in registers and LUTs used, and a 2.8x increase in clock frequency and 16x higher throughput than the floating-point implementation on the same FPGA. The throughput results are shown in I[1]

TABLE I
HOG DETECTION THROUGHPUT COMPARISON.

	CPU	IPP	FPGA-fp	GPU	FPGA-fix13
FPS	0.993	1.138	2.314	13.397	36.496
Speed-up	1.000	1.146	2.331	13.497	36.770

***Index Terms*—Computer vision; fixed-point; pedestrian detection; histogram of oriented gradients;**

REFERENCES

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition (CVPR). IEEE Conf. on*, vol. 1, 2005, pp. 886–893.

[2] R. Kadota, H. Sugano, M. Hiromoto, H. Ochi, R. Miyamoto, and Y. Nakamura, "Hardware architecture for HOG feature extraction," in *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 5th Int. Conf. on*, 2009, pp. 1330–1333.

[3] C. Blair, N. Robertson, and D. Hume, "Characterizing a heterogeneous system for person detection in video using histograms of oriented gradients: Power versus speed versus accuracy," *Emerging and Selected Topics in Circuits and Systems, IEEE J. on*, vol. 3, no. 2, 2013.

[4] K. Negi, K. Dohi, Y. Shibata, and K. Oguri, "Deep pipelined one-chip FPGA implementation of a real-time image-based human detection algorithm," in *Field-Programmable Technology (FPT), Int. Conf. on*, 2011, pp. 1–8.

[5] M. Enzweiler and D. Gavrila, "Monocular pedestrian detection: Survey and experiments," *Pattern Analysis and Machine Intelligence, IEEE Trans. on*, vol. 31, no. 12, pp. 2179–2195, 2009.

[6] C. Wojek, S. Walk, and B. Schiele, "Multi-cue onboard pedestrian detection," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2009, pp. 794–801.

[7] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2009, pp. 304–311.

[8] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "A mobile vision system for robust multi-person tracking," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on*, 2008, pp. 1–8.

[9] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *Pattern Analysis and Machine Intelligence, IEEE Trans. on*, vol. 34, no. 4, pp. 743–761, 2012.

[1] FPGA-fp refers to the floating-point implementation of HOG on the FPGA.