IEEE Asian Solid-State Circuits Conference
November 5 - 7, 2018/Tainan, Taiwan

Hardware Architecture for Fast General Object Detection using Aggregated Channel Features

Koichi Mitsunari, Jaehoon Yu, and Masanori Hashimoto
Graduate School of Information Science and Technology, Osaka University
Email: {k-mitunr,yu.jaehoon,hasimoto}@ist.osaka-u.ac.jp

***Abstract*—For embedded system applications, high detection accuracy and fast detection must be achieved within a limited power budget. This paper proposes an embedded system-oriented hardware accelerator for object detection with aggregated channel features (ACF). The proposed accelerator consists of hardware architectures dedicated for HOG features, quantization, and boosted decision trees, and they contribute to 2006X speed-up and 601X memory reduction. Our FPGA implementation result shows that the proposed accelerator can detect pedestrians at 170 fps for Full HD images, and 6-class traffic objects at 78 fps for Full HD images.**

I. Introduction

The primary goal of embedded object detection systems aiming at driver assistance and autonomous robots is to achieve fast and accurate detection with low power consumption. These applications are time-critical, and missed detection can be a threat to human life. Therefore, fast and accurate detection is indispensable and a social requirement. However, accurate detection with novel algorithms, for example, rich object representations, generally demands massive computations, which prevents fast detection and low power implementation. On the other hand, early works on object detection (e.g. [1]) use simple object representation, and the accuracy required for critical applications is not satisfied. We need to construct an accurate detection system with sophisticated algorithms and develop their efficient hardware implementations for attaining low latency and power consumption.

The automatic braking system in driver assistance needs to detect objects 33 meters ahead when driving speed is 30 km/h. In this case, Full HD 60 fps image processing is essential. Here, let us compare three well-known algorithms adopted for hardware implementation; support vector machine (SVM), convolutional neural network (CNN), and aggregated channel features (ACF) [2]. SVM is a linear classifier, and it is suitable for parallel implementation due to its simple structure [1]. However, it suffers from low detection accuracy, and it is not suitable for critical applications. CNN is drawing a lot of attention since it achieves high detection accuracy. However, due to its inherent tremendous amount of computation, low power consumption and fast detection are hardly achievable. Recently, CNN hardware is widely studied (e.g. [3]), but an implementation that satisfies high accuracy, low latency, and low power consumption is not presented yet. ACF achieves good detection accuracy. ACF uses a boosted decision tree (BDT) classifier, which requires a small amount of computation, and achieves fast detection in software implementation.

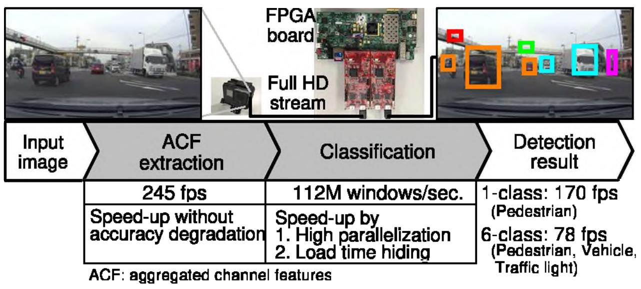


Fig. 1. Proposed Object Detection System Overview.

TABLE I
Problems in ACF Hardware Implementation.

	Memory	Area	Speed
HOG feature extraction		✗	
Feature representation	✗	✗	
Parallel classification using BDT			✗

Thanks to this, [4] achieves the fast classification of 480p30 even with a serial hardware implementation. However, its parallel hardware implementation pursuing higher throughput is not straightforward since the memory access depends on input data, and it prevents parallel implementation.

This work uses ACF as a baseline to exploit its reasonably high accuracy and low computational cost. For enhancing throughput and minimizing hardware cost, we have performed algorithm-hardware co-optimization and improved the compatibility of ACF with the hardware implementation [5]–[7]. This paper proposes a general object detection system shown in Fig. 1 and presents its FPGA implementation. ACF extraction is speeded up by adopting a hardware-oriented feature descriptor which extracts equivalent information in a small amount of computation [5], and 245 fps is achievable. BDT classification is speeded by parallel implementation and hiding load time of coefficients [7], and 112M windows/sec is attained. In addition, a quantization method which is robust to accuracy degradation [6] is adopted for memory saving and power reduction. Consequently, the proposed system can detect multi-objects of pedestrians, vehicles, and traffic signals in 1080p60, which satisfies the above-mentioned requirement for the automatic braking system.

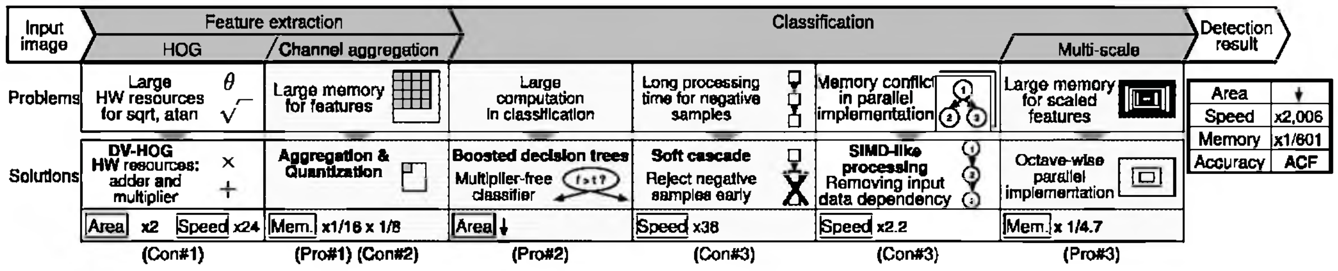

Input image
Feature extraction
HOG
Channel aggregation
Classification
Multi-scale
Detection result
Problems
Large HW resources for sqrt, atan
Large memory for features
Large computation in classification
Long processing time for negative samples
Memory conflict in parallel implementation
Large memory for scaled features
Solutions
DV-HOG HW resources: adder and multiplier
Aggregation & Quantization
Boosted decision trees
Multiplier-free classifier
Soft cascade
Reject negative samples early
SIMD-like processing
Removing input data dependency
Octave-wise parallel implementation
Area x2 Speed x24
Mem. x1/16 x 1/8
Area ↓
Speed x38
Speed x2.2
Mem. x 1/4.7
(Con#1)
(Pro#1) (Con#2)
(Pro#2)
(Con#3)
(Con#3)
(Pro#3)
Area ↓
Speed x2,006
Memory x1/601
Accuracy ACF

Fig. 2. Problems and Solutions for ACF Hardware Implementation.

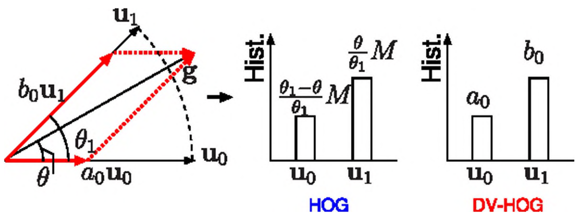


Fig. 3. HOG Feature Extraction.

II. Problems of ACF Hardware Implementation

ACF is originally developed for software implementation, and the good accuracy with small computational cost is demonstrated in [2]. The following discusses the compatibility of ACF with hardware implementation and points out problems to be addressed for high-throughput ACF implementation.

ACF has high affinity with the hardware implementation regarding the following three points. In feature extraction, ACF aggregates feature maps by 4x4, and it reduces the required memory capacity to one-sixteenth (Pro#1). In the classification step, the BDT classifier does not require multipliers, and then the necessary hardware resource is small (Pro#2), where BDT classifier consists of multiple decision trees, and at each tree, the algorithm selects a leaf node by recursive comparison between a threshold and a feature from the root node. Also, BDT can use soft cascade, which rejects negative samples early, for fast detection (Pro#3) since a cascade structure represents BDT.

However, ACF has incompatibilities with hardware implementation, which are listed in TABLE I. First, we need to extract HOG features from an input image, but this feature extraction process includes expensive computations such as trigonometric function and square root, which demands large hardware resource (Con#1). Second, a large memory is necessary to store features during feature extraction (Con#2) and classification. Finally, as explained, the parallel hardware implementation of BDT is difficult (Con#3) since the memory access patterns depend on input data, and straightforward memory segmentation cannot avoid memory access collision. We need to resolve these incompatibilities.

III. Proposed Hardware Accelerator Architecture

This section presents the proposed architecture for ACF-based object detection. Fig. 2 summarizes the problems of object detection accelerator and their solutions. Each solution contributes to at least either of area reduction, speed-up, or memory reduction. The advantages of ACF (Pro#1) to (Pro#3) provide memory reduction in channel aggregation, area reduction and fast classification in classification, respectively. (Con#1) to (Con#3), on the other hand, are resolved by the proposed architecture, and the following subsections explain the proposed hardware solutions one by one. As a result, the proposed accelerator achieves area reduction, 2,006-times speed-up, and memory reduction to 1/601 while keeping the detection accuracy almost identical to the original software ACF implementation.

A. Decomposed Vector Histograms of Oriented Gradients (DV-HOG)

HOG feature extraction calculates the magnitude and orientation of edges in an input image and generates their histogram. Gradient vector g $\in \mathbb{R}^2$ is calculated by differentiating the input image along x and y-axes, and its magnitude and angle are defined as

$$M = \sqrt{g_x^2 + g_y^2}, \quad \theta = \tan^{-1}\left(\frac{g_y}{g_x}\right). \tag{1}$$

Histogram bins are linearly spread from 0 to π radian. Given g, it votes M to adjacent bins using interpolation in terms of angle as shown in Fig. 3. HOG extraction includes the trigonometric function and square root calculation, and it requires a large amount of hardware resource.

To solve the problem, [5] proposes a complex computation-free interpolation based on vector decomposition as shown in Fig. 3. This DV-HOG regards g as the weighted sum of adjacent unit vectors and adopts L1 norm for magnitude calculation. DV-HOG is calculated only with multiplication with constant and addition, and it reduces the circuit area to 1/12 without accuracy degradation compared with interpolation in terms of angle.

B. Quantization for Decision Tree Classifiers

Classification using decision trees compares a feature and a threshold at each decision node. To reduce the memory requirement for decision tree classifiers, [6] proposed a quantization method focusing on the classifier's threshold range. Namely, child node selection in BDT is based on the comparison result between feature and threshold values, and the difference between them is not used for classification.

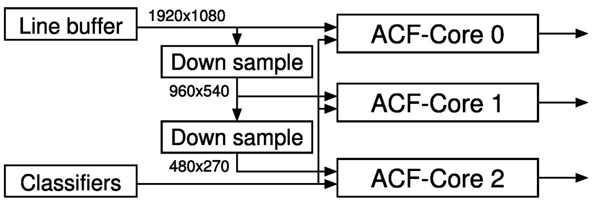
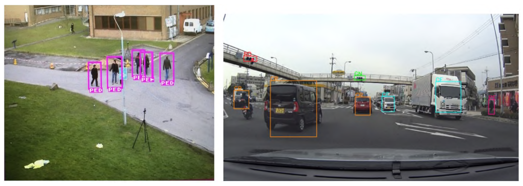


Fig. 4. Multi-scale ACF Hardware Architecture.



(a) Pedestrian detection (b) Traffic object detection

Fig. 5. Detection Results.

TABLE II
PARAMETERS USED IN THE EVALUATION.

Module		Parameter	Value
Feature extraction		Parallelism	24
		Bit-width	4bit
Classification	ACF-Core0	Parallelism	8x16x8
	ACF-Core1		8x4x8
	ACF-Core2		8x2x8

Therefore, fine quantization is applied only to the range of threshold values for memory saving. [6] reports that if 2% accuracy degradation on INRIA Person Dataset is allowed, the numerical precision is reduced from 32bit to 2bit, which indicates that memory requirement is reduced to one-sixteenth.

C. Hardware Architecture for BDT Classifier

Classification using a BDT classifier includes complicated memory access because selected decision nodes depend on input data. Thus, parallel processing is difficult due to memory access conflict. To solve the problem, [7] visits all the decision nodes and enables SIMD-like processing. When each channel has a memory bank, channel-wise parallel implementation is feasible. With the processing order scheduling and 1,024-parallel classification, the hardware can classify 350 fps multi-scale Full HD stream. Also, this method is compatible with soft cascade, and [7] reports that 33-times speed-up is obtained by soft cascade for pedestrian detection.

D. Hardware Architecture for ACF

Thanks to the above hardware solutions, the feature extraction and classification are implemented in parallel for fast detection. For detecting multiple sizes of objects, on the other hand, multi-scale detection, which scales the input image or applies classifiers of different window sizes, must be implemented. In this case, the processing time for the feature extraction is the bottleneck compared with that of classification. To reduce the feature extraction time, this architecture adopts an approach proposed by Benenson *et al.* [8]. Fig. 4 shows the overview, where the scale octave is 1/2 and classifiers dedicated for each scaled image are applied to ACF-Core 0 to 2 in parallel. In the case of Full HD image, there exist three octaves.

IV. EVALUATION ON FPGA BOARD

A. Implementation

The proposed hardware architecture is implemented on FPGA. The evaluation uses Xilinx ZC706 evaluation board and two FMC cards for HDMI input and output. The FPGA board has programmable logic (PL) and ARM core, and the ACF-HW module is implemented at register transfer level using Verilog HDL in PL. To handle the 1080p60 input stream and overlay the bounding boxes on the output stream, we used the Vivado IPs of video input, video output, and video timing controller. The trained classifier is stored in the SD card, and loaded from software running on the ARM core.

TABLE II shows the parameters used in the implementation, and the target frequency for the ACF-HW module is 100MHz. TABLE III shows the resource utilization, which shows the balanced usage of resources. We can see that DV-HOG modules occupy 8% of LUTs in total. It should be noted, on the other hand, that the original computation of Eq. (1) requires 12x resources, which makes a single FPGA implementation infeasible.

B. Object Detection Performance Evaluation

The processing performance is evaluated on pedestrian detection and traffic object detection. The evaluation using pedestrian detection aims to compare the performance with the related approaches. In traffic object detection, multi-class classification is evaluated considering practical applications.

1) Pedestrian Detection: The evaluation uses twelve classifiers whose window size ranges from 48x96 to 92x184, and each classifier consists of 2,048 depth-two decision trees. Training process uses INRIA Person Dataset [10]. Fig. 5(a) shows the detection result. For quantitative analysis, software simulation is used to count clock cycles for each step. The result shows that feature extraction and classification consume 4.08 and 1.80 ($= 0.15 \times 12$(scales)) milliseconds, respectively. Consequently, the proposed accelerator can process 170 fps of Full HD. TABLE IV shows the processing performance comparison. [1] and [9] achieve Full HD 60 fps processing. However, they suffer from the higher log-average miss rate (MR) of 46% and 20% on INRIA Person Dataset, respectively, whereas the log-average MR of the proposed architecture is 17%. For a fair comparison between ACF-based architectures, we use the processing number of windows in a second as an evaluation metric. The table indicates that the proposed accelerator achieves 57 times speed-up compared with ACF hardware [4].

2) Traffic Object Detection: As a multi-class evaluation, pedestrian, vehicle, and traffic light detection are performed on FPGA. TABLE V summarizes the classifiers. The detection candidate area is limited as shown in TABLE V to reduce the number of false positives and speed-up. Fig. 5(b) shows the

TABLE III
FPGA RESOURCE UTILIZATION.

Module	Slice		LUT (Logic)		LUT (Memory)		LUT (FF)		32Kb BRAM		DSP	
ACF-HW	124,770	(57%)	124,476	(57%)	294	(0%)	128,626	(59%)	356.5	(65%)	122	(14%)
→ ACF-Core0	70,755	(32%)	70,657	(32%)	98	(0%)	72,719	(33%)	204.5	(38%)	41	(5%)
→ ImageBuf	195	(0%)	195	(0%)	0	(0%)	601	(0%)	9	(2%)	0	(0%)
→ FeatGen	6,928	(3%)	6,831	(3%)	97	(0%)	8,215	(4%)	0	(0%)	0	(0%)
→ DV-HOG	6,168	(3%)	6,168	(3%)	0	(0%)	7,457	(3%)	0	(0%)	0	(0%)
→ AggCube	2,021	(1%)	2,021	(1%)	0	(0%)	2,878	(1%)	8.5	(2%)	40	(4%)
→ ACFCube	23,891	(11%)	23,891	(11%)	0	(0%)	24,420	(11%)	170	(31%)	0	(0%)
→ LeafCube	35,144	(16%)	35,143	(16%)	1	(0%)	36,839	(17%)	16	(3%)	0	(0%)
→ OutputBuf	39	(0%)	39	(0%)	0	(0%)	1,086	(0%)	1	(0%)	0	(0%)
→ ACF-Core1	26,127	(12%)	26,119	(12%)	98	(0%)	27,158	(12%)	84.5	(16%)	41	(5%)
→ ACF-Core2	18,859	(9%)	18,761	(9%)	98	(0%)	19,686	(9%)	64.5	(12%)	40	(4%)
Total	139,215	(64%)	137,939	(63%)	1,276	(2%)	149,129	(68%)	389	(71%)	128	(14%)

TABLE IV
DETECTION PERFORMANCE COMPARISON.

Method	Speed	Method	log-avg. MR	#win. / sec.
[1]	Full HD 60 fps	SVM	46%	6284k
[9]	Full HD 60 fps	DPM	20%	3,975k
[4]	VGA 30 fps	ACF	17%	1,972k
Ours	Full HD 170 fps	ACF	17%	112,501k

TABLE V
CLASSIFIERS FOR TRAFFIC OBJECTS.

Target	Pedestrian	Vehicle (Front, Rear)	Traffic light (Green, Yellow, Red)
Depth	2	2	2
#weak classifier	2,048	512	512
Window size	[48, 96], ..., [92, 184]	[48, 48], ..., [92, 92]	[48, 16], ..., [84, 28]
#classifier	12	12	4
Total #classifier	12	24	12
Area	Lower 2/3	Lower 2/3	Upper half

detection result, and TABLE VI summarizes the processing time. We can see soft cascade contributes to 17.6 times speed-up on average. The system can process 78 fps of full HD frames, which is enough for the driving assistance system at 30 km/h. It should be noted that the proposed accelerator does not use any domain-specific knowledge and hence it is applicable to any object detection applications, although the required fps for the driving assistance system is exemplified above.

V. CONCLUSION

Embedded object detection systems need to simultaneously achieve high detection accuracy, fast detection, and low power consumption, and its design is highly challenging. To solve the issue, this paper proposed a hardware architecture for general multi-class object detection using ACF. The proposed hardware architecture makes use of the advantages of the ACF algorithm itself and incorporates multiplier-free DV-HOG, aggressive quantization and BDT parallel computation architecture with the overall accelerator architecture. In total, the system is speed-up by 2,000 times and reduced memory to 1/600. FPGA implementation result showed that the proposed system could detect pedestrians at 170 fps for a Full HD image, and 6-class traffic objects at 78 fps for Full HD, which satisfied the requirement for the automatic braking system.

TABLE VI
CLASSIFICATION SPEED EVALUATION.

		# cycle		Speed-up
		Soft cascade		
		Off	On	
Vehicle	Front	148,677	13,626	10.9x
	Rear	153,143	15,193	10.1x
Pedestrian		575,037	14,996	38.3x
Traffic Light	Green	156,133	9,481	16.5x
	Yellow	116,308	10,716	10.9x
	Red	140,734	9,403	15.0x
Total		1,290,032	73,415	17.6x

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number JP16K16085.

REFERENCES

[1] A. Suleiman and V. Sze, "An energy-efficient hardware implementation of HOG-based object detection at 1080HD 60 fps with multi-scale support," *J. Signal Process. Syst.*, vol. 84, no. 3, pp. 325–337, Sep. 2016.

[2] P. Dollár *et al.*, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[3] R. Zhao *et al.*, "Optimizing cnn-based object detection algorithms on embedded FPGA platforms," in *Applied Reconfigurable Comput.*, 2017, pp. 255–267.

[4] H. Song *et al.*, "Hardware implementation of aggregated channel features for ADAS," in *Proc. Int. SoC Des. Conf.*, Oct. 2016, pp. 167–168.

[5] K. Mitsunari *et al.*, "Decomposed vector histograms of oriented gradients for efficient hardware implementation," IEICE Trans. Fundam. Electron., Commun. and Comput. Sci. (conditional acceptance).

[6] K. Mitsunari and J. Yu, "Influence of numerical precision on machine learning and embedded systems," in *Proc. Int. Workshop Smart Info-Media Syst. Asia*, Sep. 2016, pp. 164–169.

[7] K. Mitsunari *et al.*, "Hardware architecture for high-speed object detection using decision tree ensemble," IEICE Trans. Fundam. Electron., Commun. and Comput. Sci. (to appear).

[8] R. Benenson *et al.*, "Pedestrian detection at 100 frames per second," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. and Pattern Recognit.*, Jun. 2012, pp. 2903–2910.

[9] A. Suleiman *et al.*, "A 58.6 mW 30 frames/s real-time programmable multiobject detection accelerator with deformable parts models on full HD 1920 × 1080 videos," *IEEE J. Solid-State Circuits*, vol. 52, no. 3, pp. 844–855, Mar. 2017.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. and Pattern Recognit.*, Jun. 2005, pp. 886–893.