

An Adaptive Hardware Architecture using Quantized HOG Features for Object Detection

Ngo-Doanh Nguyen*, Duy-Hieu Bui*, Fawnizu Azmadi Hussin[†], Xuan-Tu Tran^{*‡}

*VNU Information Technology Institute, Vietnam National University, Hanoi – 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam.

[†]Department of Electrical and Electronic Engineering, Universiti Teknologi Petronas – Seri Iskandar, Perak, Malaysia.

[‡] Corresponding author's email: tutx@vnu.edu.vn

Abstract—This article presents an adaptive hardware architecture for high-performance object detection using Histogram of Oriented Gradient (HOG) features in combination with Supported Vector Machines (SVM). This architecture can adapt to various bit-width representations of HOG features by using the quantization technique. The HOG features can be represented from 8 bits to 4 bits to remove the bubble in the processing pipeline and reduce the memory footprint. As a result, the overall throughput is robustly increased as the number of bits decreases. Moreover, we propose a new cell-reused strategy to speed up the system throughput and reduce memory footprint. The proposed architecture has been implemented in TSMC 65nm technology with a maximum operating frequency of 500MHz and throughput of 3.98Gbps. The total hardware area cost is about 167KGEs and 212kb SRAMs.

Index Terms—Artificial Intelligence, Histogram of Oriented Gradient, Support Vector Machine, HOG, SVM

I. INTRODUCTION

Object detection has been a key technology in computer vision applications such as video surveillance, automobile systems and so on [1]. Generally, object detection could be solved by two main methods, including hand-crafted feature extractions and machine learning feature extraction combined with classification algorithms. For the former, numerous hand-crafted feature descriptors have been proposed, such as HOG [2], Scale-Invariant Feature Transform, Haar Transform. For the latter, machine learning feature extraction and classification has been booming with deep Convolution Neural Network (CNN), Spiking Neural Network, Recurrent Neural Network and so on. Between the two methods, the machine learning one, especially CNN, has higher accuracy, but it needs a large number of operations with millions of parameters. In contrast, hand-crafted descriptors such as HOG, combined with SVM classification, have fewer operations and fewer data dependencies [3]. Therefore, it has higher throughput and lower power consumption than the pure machine-learning-based solution.

However, for high-throughput object-detection applications, detection algorithms may be implemented into hardware to maximize the system throughput. In this case, the machine-learning feature extraction and classification cost a large amount of hardware resource and energy, which is not suitable for the lightweight embedded systems. On the other hand, HOG, represented the hand-crafted feature descriptor, along with SVM classification provides small hardware implementa-

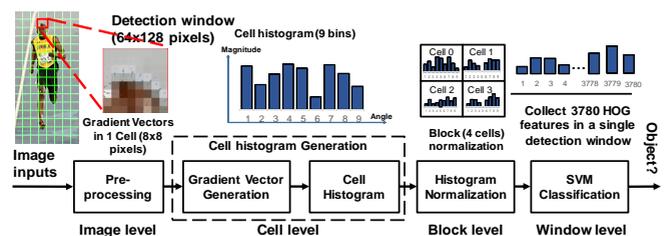


Fig. 1: HOG-SVM algorithm for object detection [4].

tion with its well-balanced trade-off among detection accuracy, system throughput and complexity [5]. As a result, many researches have been devoted to improve the efficiency of the HOG-SVM algorithm further. The steps to detect an object, for examples a human, using the HOG-SVM algorithm are described briefly in Fig. 1. The works in [4], [6] were trying to reduce the computational complexity of extracting the HOG features and the normalization process in order to increase the overall throughput. The other work in [7] show that the reusing strategy can improve the performance sharply. However, in these works, the HOG feature generation and SVM classification are still optimized separately. Our previous work in [8] overcomes this separation by proposing a hardware architecture composing of a fast, highly-parallel and low-cost HOG feature extraction combined with a parallel computation of SVM and HOG feature normalization. However, our previous work still has the bubble in the processing pipeline and does not cope with the variable bit-width for HOG features.

In this paper, we improved our work in [8] to push the throughput further by optimizing the datapath with the variable bit-width of HOG features to remove the bubble in the processing pipeline and propose a new reusing strategy. Our main contributions are listed as follows:

- This hardware architecture can support multiple-bit-width computations of the quantized HOG features by using a variable bit-width Sequential Multiply-Accumulate (SMAC).
- A new data-reuse strategy for window strides is proposed to save memory footprint and fasten SVM classification.
- This accelerator can perform at a maximum frequency of 500MHz and throughput of 240fps for Full-HD resolution with 4-bit quantized HOG features using TSMC 65nm

technology. The hardware area is only about 167KGEs and 212Kb SRAMs in a 4-bit HOG feature configuration.

The rest of this paper is organized as follows. Section II presents the related works. Section III describes our proposed hardware architecture to boost the throughput of HOG-SVM. Then, Section IV presents our hardware implementation results. Finally, there are some conclusions and perspectives in Section V.

II. STATE OF THE ART

The HOG-SVM algorithm essentially describes the detection windows using HOG feature descriptors and then applies SVM classification to detect the object. However, the original feature extraction contains many complicated arithmetic functions such as inverse tangent in histogram generation, square root, and division in the normalization step. As a result, HOG-SVM hardware implementations cost many logic gates for these functions and require huge memory for storing HOG features.

To overcome these challenges, many studies have focused on optimizing HOG feature extraction and SVM classification for hardware implementation for object detection. For instance, An *et al.* in [6] applied the approximation of trigonometric functions by converting them into the comparison among angles to avoid inverse tangent calculation for faster feature extractions. Another approximate method is to modify the square and square root computations implemented in the feature normalization step, as Mizuno *et al.* in [7]. Although those approximate computations can effectively boost the throughput and reduce hardware complexity, it reduces the accuracy of the extracted HOG features.

Another approach to improve the throughput is to apply a data reuse strategy as in [7]. By avoiding the repeated computations of the overlapped cells, the total number of operations in one detection window sharply reduces. Besides, parallelism can be applied for high-speed applications. For example, Suleiman *et al.* in [4] utilized 3 HOG-SVM detectors for multi-scale support. Our previous work in [8] also used multiple MAC modules to fasten SVM calculation.

Furthermore, some researches, as in [4], [9], proposed a new approach that performs pre-processing at the pixel level, which reduces a large number of operations and power consumption but still maintains acceptable accuracy. For example, Suleiman *et al.* in [4] presented a pre-processing technique, which uses the gradient function as a filter to reduce the input bit width before calculating the histogram. Furthermore, Young *et al.* in [9] used a converter to change the linear gradients into logarithmic gradients, which only need 2.75 bits on average to represent the HOG features. Consequently, many complicated calculations can be simplified. For instance, multiplication, division, and square in the original HOG feature extraction can turn into addition and subtraction in the logarithmic domain.

Although the previous works have investigated various aspects of hardware implementations of the HOG-SVM algorithm, the SVM classification can only start when it obtains all the normalized HOG features. This means that the throughput

of the hardware system is limited by its data dependency. Our previous work in [8] could calculate HOG features and SVM classification in parallel. However, it still has some drawbacks. The previously proposed architecture contains the bubbles in the processing pipeline, which can be further improved. In addition, it does not consider the possibility of applying the reduced-bit-width HOG feature extraction and the following processing. In this work, we reuse our previous proposed architecture in [8] and apply quantization to the HOG feature to further boost the system throughput. We also propose a new data-reuse strategy for non-square window detection. The details of our proposed architecture are explained in the following section.

III. PROPOSED HARDWARE ARCHITECTURE

Our hardware architecture can support multiple object detection by modifying the size of the detection window and the SVM-trained weights. However, for easier explanation and evaluation, we chose the pedestrian as the object detection with two datasets, including the INRIA dataset [2] and the TUD dataset [10]. The overview of our proposed architecture is described in Fig. 2.

Fig. 2 shows eight pix2bin modules are utilized in parallel to generate bin histograms before quantization. After accumulating all the bins into one cell histogram containing 9 orientation bins, our design performs quantization by keeping only the most significant bits of the cell histogram. The number of bits representing the cell histogram can be configurable at design time. The quantized cell histograms are used for accelerating the SVM classification and normalization process.

A. Quantization of Non-Normalized HOG Feature

To reduce the hardware area and increase the throughput, the HOG features can be quantized by discarding the least significant bits. Consequently, only the most significant bits are kept for further processing. In our previous work, we used the non-quantized HOG features. In this work, we use the quantized HOG features to increase the system throughput in $L2$ normalization and SVM classification. Quantized HOG features help remove the bubble in the processing pipeline leading to a $1.7\times$ improvement in throughput compared with our previous work. Fig. 3 shows the new data pipeline processing for 8-bit and 4-bit HOG features. Obviously, the bubble in the data processing pipeline is completely removed, which increases the system throughput.

B. Data Reuse and Pipeline Strategy

Data reuse strategy and pipeline architecture are important for high-speed designs. In this work, we firstly reuse the generated cell histograms by storing 128×9 quantized features in a buffer. At this point, quantized cell features are arranged into two data paths: new-cell calculations and overlapped-cell calculations [8].

As shown in Fig. 4, the detection window could move in four directions: up, down, right and left. 8 new cells and 120 overlapped cells need calculating in the up/down

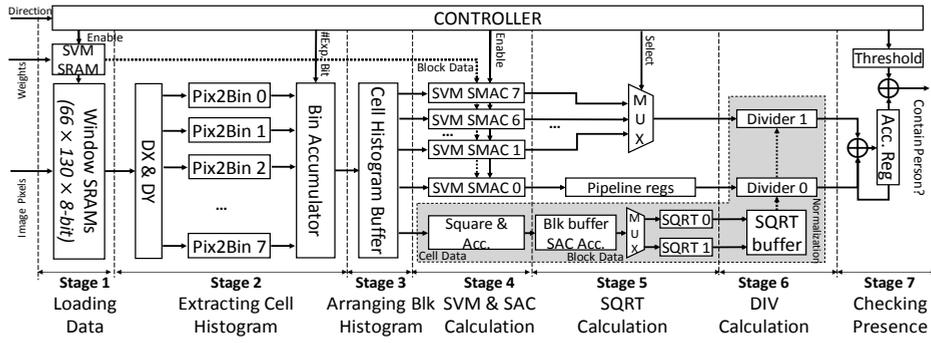
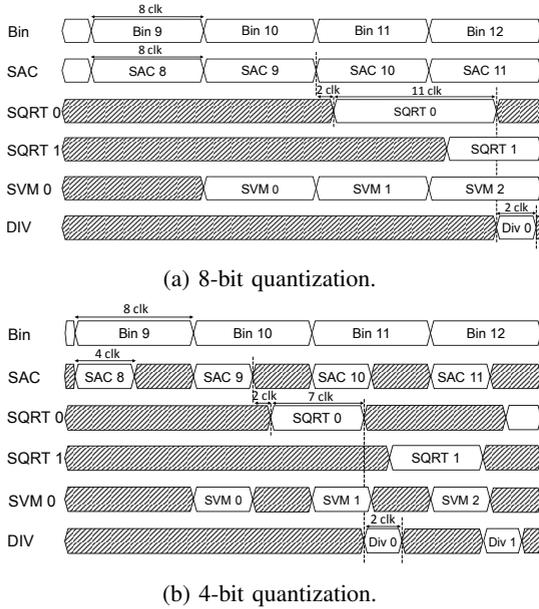


Fig. 2: Proposed Hardware architecture.



(a) 8-bit quantization.

(b) 4-bit quantization.

Fig. 3: Data pipeline of the proposed architecture for quantized cell histogram.

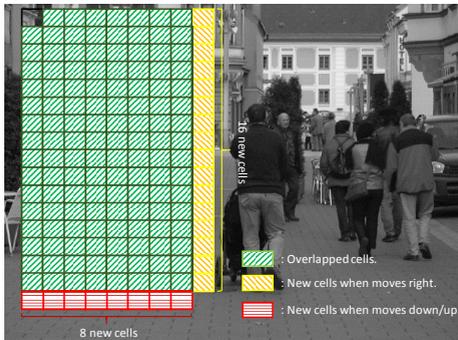


Fig. 4: The cell organization of the detection window at the frame level.

movement instead of 16 new cells and 112 overlapped cells in the right/left movement. Therefore, our architecture uses the majority of up/down movement and avoids the right/left

TABLE I: Hardware Implementation Result using 65nm TSMC Technology.

	This work		[4]	[6]	[8]
	4-bit	8-bit			
Technology	65nm TSMC	45nm SOI	65nm SOTB	65nm TSMC	
Feature	HOG	HOG	HOG-Haar	HOG	
Resolution	Full HD	Full HD	Full HD	Full HD	
Hardware Area	167KGEs	195KGEs	490KGEs	500KGEs*	145KGEs
Memory	0.212Mbits		0.538Mbit	0.602Mbits	0.242Mbits
Frequency	500MHz		270MHz	200MHz	500MHz
Power Consumption	151mW	195mW	45.3mW	75.48mW	-
Frame Rate	240fps	139fps	60fps	30fps	139fps
Energy Efficiency	304p/pix.	677p/pix.	364p/pix.	1521p/pix.	-

* The HOG core area is calculated from the original paper based on the best of our knowledge.

movement to maximize the data reuse and increase the system throughput. For example, to completely scan a Full-HD image containing 233×120 windows of 128×64 pixels, it will need 232×120 right/left movements along with 120 movements in the down direction. In contrast, it needs 233×119 movements in the up and down direction with 233 movements in the right direction. Consequently, our design can improve the data reuse by saving about $27,000 \times n$ cycles by choosing the up/down movements and minimizing the right/left movement.

IV. EVALUATION

Our proposed architecture has been implemented in the TSMC 65nm standard cell library, SRAM model from ARM and it has been verified by the dataset from [2] and [10]. Our trained weights are constructed based on over 3,000 images of 96×160 pixels from [2]. We create multiple trained weights to achieve the best accuracy of each quantized HOG feature. They are then tested on three different scales of 288 images from [2] and 250 images from [10].

A. Throughput

As shown in Table I, the maximum throughput of this proposed hardware implementation is about 240 fps at Full

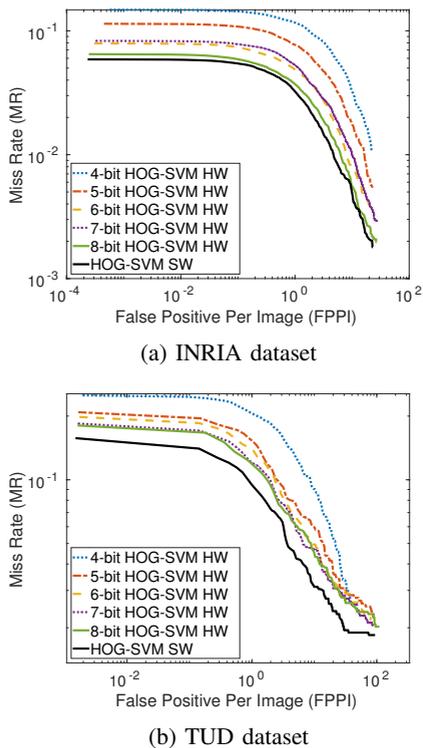


Fig. 5: Performance comparison between multiple bit width of HOG features on the INRIA and TUD dataset.

HD (3.98Gbps) when the number of quantized cell histograms is 4 bits. Compared with other works, our design achieves the highest frequency at 500MHz and the highest framerate at 240fps for Full-HD images. The proposed architecture can perform $1.7\times$ times faster at the same frequency when compared with our previous work in [8], which does not apply these optimization techniques. On the other hand, our design throughput is $4\times$ and $10\times$ faster than the previous works in [4] and [6], respectively.

B. Hardware Resource

As shown in Table I, our hardware area for 4-bit quantized HOG features costs around 167KGEs and has an average power consumption of 151mW with 212Kb SRAMs. Compared with our previous work, we have improved the system throughput at the cost of storing more features in registers leading to an increase of 50KGEs in the 8-bit version. Our proposed design is the second smallest design with the highest energy efficiency among the other works.

C. Accuracy

To evaluate the accuracy of the proposed architecture, we first obtain the suitable trained weights with over 3,000 images from the dataset in [2] by modifying the loss factor in the loss function of the SVM algorithm. This process is operated repeatedly for each version of quantized-HOG-feature bit width. Those trained weights are then loaded into our hardware design along with the positive and negative images of the two

datasets with different scales. As a result, the detections of our hardware implementation with multiple versions of the quantized feature's bit width are shown in Fig. 5a and Fig. 5b.

In Fig. 5, the miss rate of the implementation results among multiple versions of HOG features steadily increases as the degradation of HOG features. For both datasets, the miss rate of our hardware architecture with 8-bit HOG features is close to the software one at a False Positive Per Image (FPPI) of 20. At the same FPPI, the 4-bit version has a $5.55\times$ increase in the miss rate with an error rate of about 1.5% for the Inria dataset and a $1.29\times$ increase in the miss rate with TUD dataset. The proposed architecture can adapt to different throughputs at the cost of an increase in the classification error.

V. CONCLUSIONS

Object detection has a wide range of applications such as robotics, automobile, and video surveillance. One of the efficient methods to perform object detection for lightweight embedded systems is HOG-SVM. However, HOG-SVM, with its complexity and data dependencies, limits its throughput for high-speed hardware implementation. In this paper, we proposed a hardware architecture supporting quantized HOG features to remove the bubble in the previous design combined with a new data-reused strategy. The proposed hardware architecture can run at the maximum frequency of 500MHz in TSMC 65nm technology with a throughput of 240fps for Full-HD resolution using 4-bit HOG features.

ACKNOWLEDGEMENT

This work is partly supported by Vietnam National University, Hanoi (VNU) through research project "Investigate and develop a secure IoT platform".

REFERENCES

- [1] C. Bila, F. Sivrikaya, M. A. Khan, and S. Albayrak, "Vehicles of the future: A survey of research on safety issues," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1046–1065, 2017.
- [2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE-CVPR*, vol. 1, June 2005, pp. 886–893.
- [3] A. Suleiman, Y.-H. Chen, J. Emer, and V. Sze, "Towards closing the energy gap between hog and cnn features for embedded vision," in *ISCAS*, May 2017, pp. 1–4.
- [4] A. Suleiman and V. Sze, "Energy-efficient hog-based object detection at 1080hd 60 fps with multi-scale support," in *IEEE-SiPS*, Oct 2014.
- [5] W. Zhou, S. Gao, L. Zhang, and X. Lou, "Histogram of oriented gradients feature extraction from raw bayer pattern images," *IEEE TCASII*, vol. 67, no. 5, pp. 946–950, 2020.
- [6] F. An, X. Zhang, A. Luo, L. Chen, and H. J. Mattausch, "A hardware architecture for cell-based feature-extraction and classification using dual-feature space," *IEEE TCSVT*, vol. 28, pp. 3086–3098, Oct 2018.
- [7] K. Mizuno, Y. Terachi, K. Takagi, S. Izumi, H. Kawaguchi, and M. Yoshimoto, "Architectural study of hog feature extraction processor for real-time object detection," in *IEEE SIPS*, 2012, pp. 197–202.
- [8] N. Nguyen, D. Bui, and X. Tran, "A novel hardware architecture for human detection using hog-svm co-optimization," in *2019 IEEE APCCAS*, 2019, pp. 33–36.
- [9] C. Young, A. Omid-Zohoor, P. Lajevardi, and B. Murmann, "A data-compressive 1.5/2.75-bit log-gradient qvga image sensor with multi-scale readout for always-on object detection," *IEEE JSSC*, vol. 54, no. 11, pp. 2932–2946, 2019.
- [10] C. Wojek, S. Walk, and B. Schiele, "Multi-cue onboard pedestrian detection," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 794–801.