Real-Time Multi-Scale Pedestrian Detection for Driver Assistance Systems

Maryam Hemmati
Dept. of Electrical & Computer Engineering
University of Auckland
Auckland, New Zealand
m.hemmati@auckland.ac.nz

Morteza Biglari-Abhari
Dept. of Electrical & Computer Engineering
University of Auckland
Auckland, New Zealand
m.abhari@auckland.ac.nz

Smail Niar
LAMIH/CNRS
University of Valenciennes and Hainaut-Cambresis
Valenciennes, France
smail.niar@univ-valenciennes.fr

Stevan Berber
Dept. of Electrical & Computer Engineering
University of Auckland
Auckland, New Zealand
s.berber@auckland.ac.nz

ABSTRACT

Pedestrian detection is one of the most challenging and vital tasks of driver assistance systems (DAS). Among several algorithms developed for human detection, histogram of oriented gradients (HOG) followed by support vector machine (SVM) has shown the most promising results. This paper presents a hardware accelerator for real-time pedestrian detection at different scales to fulfill the real-time requirements of DAS. It proposes an algorithmic modification to the conventional multi-scale object detection by means of HOG+SVM to increase the throughput and maintain the accuracy reasonably high. Our hardware accelerator detects pedestrians at the rate of 60 fps for HDTV (1080x1920) frame.

CCS CONCEPTS

• **Computer systems organization → Embedded hardware**;

KEYWORDS

Pedestrian detection; HOG; SVM; multi-scale; hardware accelerator; real-time; FPGA

ACM Reference format:
Maryam Hemmati, Morteza Biglari-Abhari, Smail Niar, and Stevan Berber. 2017. Real-Time Multi-Scale Pedestrian Detection for Driver Assistance Systems. In *Proceedings of ACM/EDAC/IEEE Design Automation Conference, Austin, TX, USA, June 18-22, 2017 (DAC '17)*, 6 pages.
DOI: http://dx.doi.org/10.1145/3061639.3062308

1 INTRODUCTION

Driver assistance systems (DAS) are getting more popular in the modern cars to provide safer travels for people, and pedestrian detection as one of the major and crucial tasks of DAS, has attracted lots of attention in recent years. The accurate detection as well as the hard real-time requirements of the task has kept it yet in the center of attention of many researchers and automotive companies. While accurate detection requires more sophisticated and complicated algorithms and consequently higher computational complexity, every second of the time given to the driver to react in critical situations becomes vital and decisive.

The perception-brake reaction time (PRT) which is defined as the amount of time since the hazardous signal has occurred until when the driver has reacted to that has a wide varying value for different drivers and different environmental conditions. Parameters that may affect PRT are the weather, visual acuity, driver expectation, and driver alertness which in turn depends on the driver character, age, and cognitive load [8]. PRT could range between 0.7 second to around 1.5 seconds or more; however, for ease of use we may consider the nominal value of 1.5 seconds for it [8]. Other than driver reaction, it takes a while for the vehicle to get fully stopped since the brake has been pushed, depending on the initial speed and deceleration as well as the road friction. Considering a value of $6.5m/s^2$ for the vehicle deceleration, the braking distance in case of traveling at the speed of 50Km/h would be 14.84m. In the case of 70Km/h the braking distance increases to 29.16m. The total stopping distance of the vehicle is the sum of braking distance and perception reaction distance. Assuming 1.5s for the perception reaction time, the total stopping distance at these two speeds would be 35.68m and 58.23m respectively. Therefore the DAS should be capable of detecting objects within around 20m to 60m of distance from the vehicle while the detection time should be minimized as much as possible.

Extracting the HOG features from the image and applying SVM classifier to them has shown promising result for detection of human and has been used in several works either in its original or modified version. While the algorithm has shown competent detection results, the processing time and required computational load is yet an issue for real-time embedded applications.

We present a hardware accelerator for real-time pedestrian detection by employing the HOG feature extractor presented in [10], followed by our novel feature scaling module and several instances of SVM classifiers to detect the human presence at different sizes and distances. Parallel and deep pipelined architecture of the HOG feature extractor and SVM classifier has led to high throughput while minimizing the memory utilization for the intermediate result

DAC '17, Austin, TX, USA

DOI: http://dx.doi.org/10.1145/3061639.3062308

storage between the extractor and classifier stages. We introduce new method of HOG feature scaling to generate feature pyramids instead of image pyramid for fast scale variant object detection with reasonable accuracy. Employing several instances of the SVM classifier could provide real-time multiple object detection capability which is highly demanded in applications such as driver assistance systems.

The rest of the paper is organized as follows. A review of state of the art implementations for pedestrian detection is provided in Section 2. An overview of HOG and SVM algorithms is presented in Section 3. Description of our own method in using HOG feature pyramid instead of image pyramid and supporting mathematical analysis are explained in Section 4. Section 5 explains the hardware implementation. Concluding remarks are discussed in Section 6.

2 RELATED WORKS

Several works have been done in the area of pedestrian detection to improve the detection efficiency by introducing new or improved versions of current methods in different stages of detection, i.e. feature extraction, and classification [9, 12, 13, 19]. Some researchers have introduced new pedestrian datasets or an extension to the current available datasets to cover more human poses and environmental conditions resulting in more robust trained model for their classifier [6]. Some others have focused on improving the feature extraction algorithms to achieve higher detection accuracy [4, 16]. Among several available detection algorithms, combination of HOG feature extractor and SVM classifier has shown promising results and has been widely used in different implementations of pedestrian detection [9, 15]. It has also been employed in detection of other object classes such as vehicles and has shown reasonably accurate detection results compared to the other vehicle detection algorithms widely used [17].

A parallel array processing engine for a generic SVM classifier is implemented on FPGA [11] to handle multi class object detection in real time applications. Their presented array architecture provides scalable array size based on hardware demands so that it could be adapted to specific application. Mishra et al [14] implemented a hardware accelerator for simultaneous image resizing to support scale-invariant object detection while improving the execution time of detection. Hahnle et al. [9] employed SVM classifier with the sliding window over the original and resized input images to perform pedestrian detection at multiple scales. They considered eighteen different scales of pedestrian and managed to satisfy real time constraints by introducing the paradigm of time multiplexing over six instances of HOG feature extractor and SVM classifier. Their time-multiplex approach is implemented by reconfiguration of scaling modules.

Dollar et al. [4] introduced fast feature pyramids for object detection where multi-resolution image features are extracted from the feature descriptor by extrapolation for nearby scales. Checking the results of both pedestrian detection and general object detection for various available datasets, they have shown that the approximation does not sacrifice performance [4]. Their approach reduced the required image resizing scales by a factor of 10 resulting in lower computation load during feature extraction. In [5] Dollar et al. have approximated the feature response in nearby scales to address the

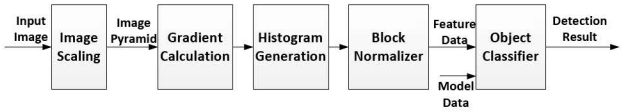

Figure 1: Detection block diagram based on HOG feature extractor and SVM classifier.

bottleneck of image pyramid construction in object detection algorithms. They generated trained SVM models in various scales and applied them to windows of different sizes during the classification. In [1] different scales are handled during the training of SVM classifier, transferring the computation from test time to training time. They achieved 135 fps detection by reverting the approach of the detector introduced in [5] to avoid input image resizing at different scales.

We present a hardware accelerator of multi-scale pedestrian detector implemented on a Zynq FPGA development board for real-time applications. Multi-scale detection is achieved by scaling the HOG features instead of the original image, which has been done in previous works to increase the detection speed for real-time applications. Generating HOG feature pyramids by using HOG features at original scale is based on the results generated by MATLAB simulations. Our design is capable of detecting human in high resolution HDTV image at two different scales and at the rate of 60fps to fulfill the requirements of real-time pedestrian detection in driver assistance system.

3 DETECTION ALGORITHM

The principle of object detection includes two different stages of feature extraction and object classification where specific features of an image are extracted at the first step, and based on the calculated features the classifier decides whether the object belongs to the specific class or not. Figure 1 shows a block diagram of object detection using HOG feature extractor followed by SVM classifier. In this method, the HOG features are first calculated and then together with the model data, which is trained for specific object class, are fed to the classifier. Initially, the gradients within small divided parts of the image, called cells, are computed. Then the orientation histogram would be generated based on the resulted gradients within each cell.

3.1 HOG Feature Extraction

HOG features are defined based on the fact that the local appearance of an object can be described by its local intensity gradient distribution so that even without precise information of the intensity and its gradient for each pixel, the object shape could be characterized well enough for the purpose of detection [2, 3]. Calculating HOG features incorporates dividing the input image into small parts called cells, normally 8x8 pixels as shown in Figure 2 and then calculating the gradient histogram for each cell. These histograms are then normalized in the next stage to suppress the effect of different local brightness and contrast in the image [2, 3]. Histogram of orientation for each cell in the image is generated based on the value of $m(x, y)$ and $\theta(x, y)$ where $f_x(x, y)$ and $f_y(x, y)$

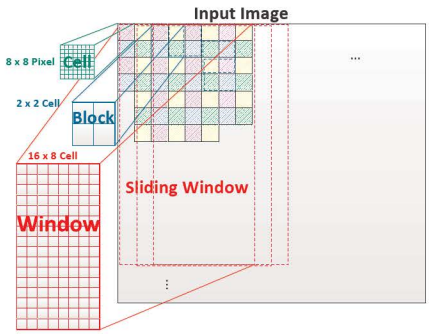

Figure 2: Cell, block, and detection window in HOG algorithm. Each cell is defined as an area of 8x8 pixels and each block includes four adjacent cells providing overlap with their neighbouring blocks. Detection window is 64x128 pixels equivalent to 8x16 cells. Sliding each window by one cell either in vertical or horizontal direction results in a new detection window.

are defined as the gradient in horizontal and vertical directions respectively. Orientation bins are defined by dividing the interval of $[0, \pi)$ evenly to the number of orientations.

$$m(x, y) = \sqrt{f_x^2(x, y) + f_y^2(x, y)} \tag{1}$$

$$\theta(x, y) = \arctan \frac{f_y(x, y)}{f_x(x, y)} \tag{2}$$

Number of orientations is assumed equal to 9 since it has shown better results in the case of human detection [3]. Two nearest bins to each gradient direction would be updated each by a score which is based on the magnitude of gradient as well as the distance of gradient angle to the edge angle of each bin [2]. The final step in HOG feature extraction is the normalization process across the group of adjacent cells defined as blocks.

3.2 SVM Classification

Once the features are extracted they could be fed to the classifier. Within the classification stage several windows of image descriptor are evaluated for the presence of specified object.

Linear SVM classifier is a binary classifier which could be trained by an object dataset including both positive and negative sets. Once classifier is trained, it constructs a hyper-plane by its support vectors which defines either a specific set of features belongs to a class of objects or not [18]. SVM classifier looks for the answer of the equation (3) in a way that w is minimized so that $E(w)$, the total hinge loss, is minimized.

$$E(w) = \frac{\lambda}{2} \|w\|^2 + \frac{1}{n} \sum_{i=1}^{n} max\{0, 1 - y_i \langle w, x \rangle\} \tag{3}$$

During the detection and at classification stage, linear SVM classifier compares the test data with model data by evaluating the equation (4) Where w is the weight vector calculated and obtained

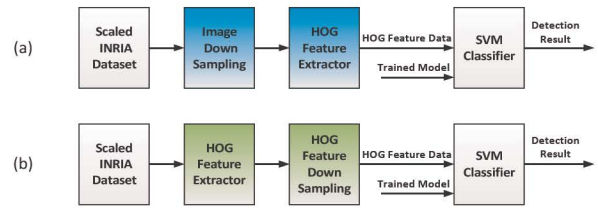

Figure 3: Setup test for two scenarios of (a) Conventional detection method (b) Proposed detection method.

during the training stage and b is the bias which is a constant value extracted from the training phase.

$$y(x) = w.x + b \tag{4}$$

During the classification stage, the dot product of weight vector with each feature vector x is calculated and added to the bias value. The resulted $y(x)$ defines whether the feature vector belongs to the specific class of objects or not by checking its sign as

$$\begin{cases} y(x) > 0 \Longrightarrow positive & (5) \\ y(x) < 0 \Longrightarrow negative & (6) \end{cases}$$

4 MULTI-SCALE DETECTION

Object detection by means of HOG feature extractor and SVM classifier includes several iterations of image scanning at different scales due to the scale invariant nature of HOG features [2, 3]. HOG feature extraction is considered as an intensively computational stage of the detection process. The computational complexity is increased when multi-scale object detection is required as image pyramid should be constructed by down-sampling the original image at several scales to fit the scaled object within the constant size detection window.

We propose a modified HOG+SVM detection method for multi-scale object detection in which the normalized HOG features are down-sampled to detect different sizes of the object within an image. By shifting the scale pyramid generation stage to the later stages after the feature extraction, the computational complexity will be reduced significantly. The INRIA person dataset has been used to verify the effect of our proposed modification. The original dataset was used to provide an SVM model for pedestrian by training a linear SVM with the extracted HOG features in LibLinear [7]. The model was then used to check the detection accuracy where 98.0375% detection rate was obtained. The original test dataset of INRIA was then up-sampled by using the scale value of 1.1 to 2 with the step size of 0.1 to generate a test dataset for human at various window sizes from 64x128 to 128x256.

The verification was done by applying the up-sampled test dataset to two different configurations of detector as shown in Figure 3. The first configuration follows the conventional method of HOG+SVM detection where the image is first resized to the detection window size and then the features are extracted and fed to the classifier. The second configuration shows our proposed method in which

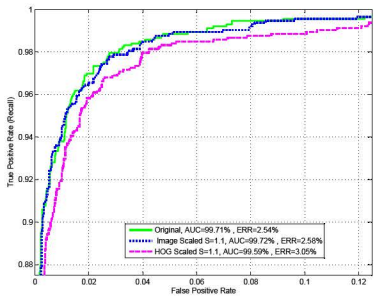

Figure 4: ROC curves for different test scenarios.

Table 1: Detection accuracy and number of true positives, and true negatives for different scales of original image and HOG feature, examined on INRIA dataset.

Scale	Accuracy		True Pos.		True Neg.	
1	98.0375%		1083		4462	
	Image	HOG	Image	HOG	Image	HOG
1.1	96.9413%	97.8076%	1102	1053	4381	4479
1.2	96.9236%	97.5778%	1100	1038	4382	4481
1.3	96.8883%	97.4187%	1103	1019	4377	4491
1.4	97.0827%	97.7192%	1102	1039	4389	4488
1.5	97.4894%	97.2419%	1093	1017	4421	4483

the HOG features are extracted at the first stage and then the features are resized to match the detection window size equal to the dimension of trained SVM model. Detection accuracy was calculated and compared for these two different methods to verify the feasibility and reliability of our proposed method. Test dataset has 1126 images that include human, called as positive test images as well as 4530 negative test images where there is no human in them. Negative test windows have been randomly sampled from INRIA negative images as advised in [3]. Results of the comparison including the detection accuracy as well as the number of correctly detected positive and negative test images are presented in Table 1.

Obtained results for the accuracy shows that at near original scales, up to the scale value of around 1.5, our proposed method outperforms the conventional method while as the scale value increases from 1.5 to higher values, down-sampled HOG features are not as promising as the resized image.

The receiver operating characteristic (ROC) curves are plotted for different test scenarios. For the purpose of comparison, the ROC curves for two different scaling methods at original size and for the scale value of 1.1 are shown in Figure 4 . The Area Under the Curve (AUC) and Equal Error Rate (EER) are shown to give an idea on how well each classifier is working. AUC which in ideal case is equal to one is considered as an indicator of the overall quality of the classifier and EER is the error rate at the point where both false positive and false negative rates are equal. False negatives are the number of test images in which the presence of the pedestrian has not been detected by the classifier while false positives are the number of images without any pedestrian which are wrongly considered to include human. The trade-off between the false positives and false negatives could be handled by varying the threshold in the classifier.

5 IMPLEMENTATION

The hardware accelerator of pedestrian detector including HOG feature extractor and SVM classifier is implemented on Zynq development board by HDL description. Multi-scale pedestrian detection is considered in the implementation due to the fact that extracted features are not scale invariant and detection of pedestrians with different size and distance is required. Generally for object detection by means of HOG features, image pyramid is generated from the input image and features are extracted for several scales followed by classification stage for each of them [2]. However, since histogram generation is the most computational intensive part of the detection chain, calculating HOG features for several scales would impose high computational load and inevitably results either in long processing time or high resource utilization based on the implementation.

Our implementation includes the scaling module which downscale the HOG features instead of resizing the original image. In this way we have escaped from the repetition of computational intensive task of histogram generation and consequently higher throughput is achieved while maintaining reasonably low resource utilization. Due to the high computational load of SVM algorithm and considering the fact that the computation should be repeated for each sliding window on the image, fully parallel and pipelined architecture is required to fulfill the requirements of real-time detection. However extracting the parallelism through the algorithms are always challenged by several constraints among them memory access bandwidth is the most notable one. The challenge of memory bandwidth would be tougher once the memory access pattern in different stages of calculation is changing. The memory storage pattern which suits the HOG feature extractor stage is not necessarily the best option for the following stage of SVM classifier since classification is applied to each window of 64x128 while histograms are generated for each row of cells in the image as the input pixels are swept horizontally. In order to leverage the parallel architecture of FPGA for both HOG feature extractor and SVM classifier in the most optimized way, we have employed specific memory storage pattern as well as specific structure for calculation units within the classifier components.

Hemmati et al. [10] introduced an efficient memory structure for the storage of normalized HOG features by dividing the cells into four different groups and storing the final results of HOG extractor in 16 different memory banks. We use a similar memory structure for our HOG extractor and reduce the storage element of NHOGMem to be used just as the middle buffer. Ensuring that our classifier is as fast as the previous HOG extractor stage we have reduced the size of NHOGMEM to store only 18 rows of cells instead of 135 rows which was used in [10].

Figure 5 shows the block diagram of implemented hardware for pedestrian detection including the HOG feature extractor and SVM classification block which itself contains the scaling module required for multi-scale detection. During the initial stage of detection, normalized HOG features are calculated and stored in a temporary storage element with specific pattern to provide the required memory access bandwidth for the next stage. When the

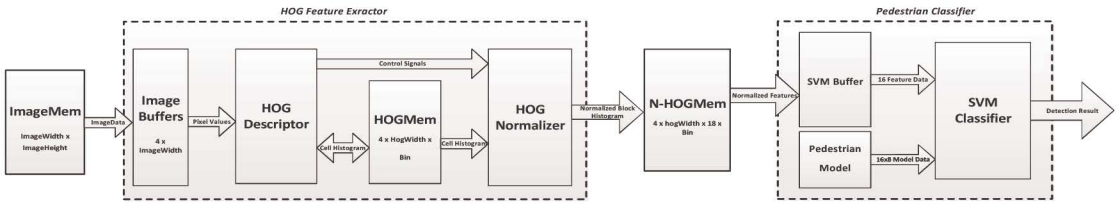

Figure 5: Block diagram of implemented hardware for pedestrian detector including HOG feature extractor and SVM classifier.

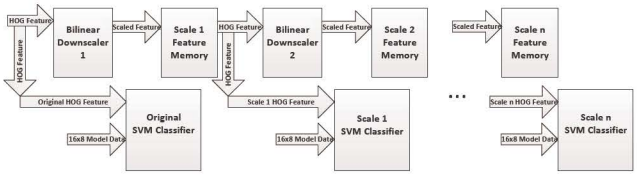

Figure 6: Multi-scale classification accomplished by a series of pipelined down-scaling modules which resize the HOG feature of prior scale. Temporary feature memories at each scale provide required data for SVM classifiers as well as next level down-scaling module.

buffered feature data are ready to be fed to the classifier, the model data would be also fed to the classifier at the same time and the dot product between each feature vector and model data would be calculated. Pedestrian model is the weight vector resulted from off-line training process for pedestrian dataset which is stored in a separate memory to be accessed by several instances of SVM classifiers, detecting the pedestrian in different scales of image. Each detection window is consisted of 16x8 blocks and each of the blocks has the feature vector of 36 elements. Parallelizing SVM classifier as well as moving the scaling block post HOG feature extraction stage has resulted in significant throughput increase.

Figure 6 shows how multi-scale classification is achieved by down-scaling the HOG features. Temporary and partial data storage and data access for several features are managed through the pipelined implementation. As discussed previously, one of the main bottlenecks in achieving high throughput is the high demand of memory access in object detection algorithms. During several stages of the processing, calculated values should be stored to be accessed for further processing in the following stages. In many cases the calculated values generated within a stage could not be sent directly to the following stage, as the next stage requires some other data before it could be able to start its function. In such scenarios using storage elements seems inevitable, however updating the memories by the end of one processing stage and reading back the data from memory in the next stage will result in delayed function of processing blocks and consequently, will reduce the throughput. It will also result in higher memory utilization in the system. We have managed the data transmission between several stages of feature scaling by employing only temporary data storage and pipelined structure as shown in Figure 7. Scaling modules are implemented by shift-and-add instead of multiplier to keep resource utilization as low as possible.

Data access in SVM classifier is also managed through pipelined structure of classifier compatible with feature memory which provides access to 16 different HOG features through 16 memory banks. However, the previous memory stage provides 16 simultaneous data; those are not the ones in one column of the window which is required to be available at the same time for the classifier. SVM classifier is capable of calculating the dot product for two block columns every 72 clock cycles by circling through four different categories of feature data groups, i.e. LU, RU, LB, and RB as introduced in [10]. It is possible to access the whole feature of two blocks through 72 cycles. Consequently, after the initial 288 cycles required for the buffer to get full, every 36 clock cycles one column of blocks is read from memory and multiplied in the trained SVM model.

Once the challenge of memory access is addressed, the parallel structure of SVM classifier could be employed efficiently. To increase the throughput as much as possible so that real-time requirements are satisfied, we have defined a deep pipelined parallel architecture for the classifier calculation stage. The importance of high throughput would be realized once maintaining the same processing speed at both stages of feature extraction and classification results to eliminate unnecessary storage elements. As the data features for one column of the window are fed to the classifier the dot product would be calculated by 16 different MAC units each responsible for multiplication and accumulation which is required in dot product. Each block in the window has 36 feature data which requires the MACBAR to calculate all of them through 36 cycles. Once one column completes the calculation the data would be piped through the next MACBAR. Consequently, once all eight MACBAR units are fed with the data, the classifier can calculate the SVM result for window every 36 cycles. Therefore one window of features is processed and evaluated by the classifier for the presence of pedestrian object. This continues until the window of image reaches to the end of the row when another 288 cycles are required to fill the SVM buffer. Consequently, the classifier can complete its job for a frame of image within 1200420 clock cycles. Considering the fact that our design is running at 125MHz, each frame of image is processed within less than 10ms. Figure 7 shows MACBAR parallel architecture, which is consisted of 16 MAC working in parallel each fed with a model data and data feature separately. Figure 8 shows the parallel and pipelined architecture of the SVM classifier, which is consisted of 8 parallel MACBAR computation unit. The

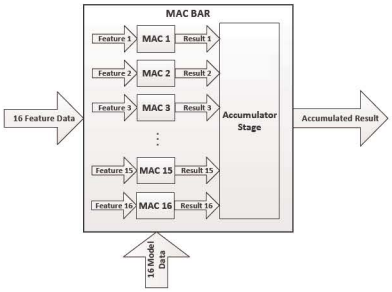

Figure 7: MACBAR parallel architecture.

Table 2: Resource utilization of hardware accelerator.

LUT	FF	LUT RAM	BRAM	DSP48	BUFG
26057	40190	383	98.5	18	1
49.61%	37.77%	2.28%	73.21%	8.18%	3.13%

feature data fed to the classifier are pipelined through eight stages calculating 8 columns of a window.

Table 2 shows a summary of resource utilization for the hardware accelerator implemented on Zynq ZC7020. Due to the memory limitations only two scales of HOG features have been considered in the classification process. However, by employing a larger device with more resources, the design could be easily extended to cover several scales for human detection which improves the accuracy by including a wider range of pedestrian size and distance. Our design running at 125 MHz, is capable of real-time detection for HDTV (1080x1920) frame at the speed of 60 fps.

6 CONCLUSION

A hardware accelerator for multi-scale pedestrian detection was presented for real-time driver assistance system. It is based on the modification made to the conventional algorithm to avoid repetitive calculations of HOG features at different scales and increase the detection speed. Generating scale pyramids of HOG features based on the calculated HOG features from the first stage reduced the computational complexity and increased the detection speed. Results show that the proposed method has not affected the detection accuracy more than 2% while reduced the computational complexity effectively. Mathematical analysis conducted in MATLAB for INRIA dataset shows that as long as down-sampling is done with the scale value of less than 1.5 the results for the modified method outperform the conventional algorithm with regards to the detection accuracy.

Within the modified algorithm, the feature extractor stage has been integrated with the scaling stage and subsequently the classifier stage. Our novel memory access approach in integration of these processing stages has maintained the high throughput rate obtained in the first processing stage. Moreover, the fully parallel architecture of SVM classifier has provided the capability of simultaneous classification for the sliding detection window within a frame. Current implementation can detect pedestrian objects at two

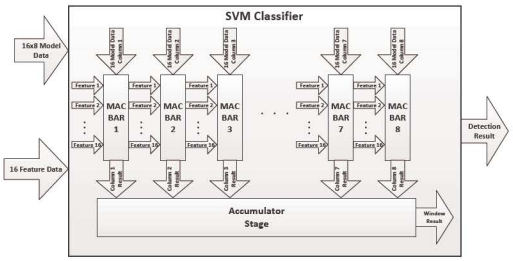

Figure 8: SVM Classifier parallel and pipelined architecture.

different scales for an HDTV frame within 16.6ms which results in feasibility of detection for 60fps HDTV stream.

REFERENCES

[1] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool. 2012. Pedestrian detection at 100 frames per second. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2903–2910.

[2] Navneet Dalal. 2006. *Finding People in Images and Videos*. Ph.D. Dissertation. Grenoble Institute of Technology, France.

[3] N. Dalal and B. Triggs. 2005. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1. 886–893 vol. 1.

[4] P. Dollar, R. Appel, S. Belongie, and P. Perona. 2014. Fast Feature Pyramids for Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 8 (Aug 2014), 1532–1545.

[5] Piotr Dollar, Serge Belongie, and Pietro Perona. 2010. The Fastest Pedestrian Detector in the West. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 68.1–68.11.

[6] P. Dollar, C. Wojek, B. Schiele, and P. Perona. 2012. Pedestrian Detection: An Evaluation of the State of the Art. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 4 (April 2012), 743–761.

[7] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *J. Mach. Learn. Res.* 9 (June 2008), 1871–1874.

[8] Marc Green. 2000. "How Long Does It Take to Stop?" Methodological Analysis of Driver Perception-Brake Times. *Transportation Human Factors* 2, 3 (2000), 195–216.

[9] M. Hahnle, F. Saxen, M. Hisung, U. Brunsmann, and K. Doll. 2013. FPGA-Based Real-Time Pedestrian Detection on High-Resolution Images. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 629–635.

[10] M. Hemmati, M. Biglari-Abhari, S. Berber, and S. Niar. 2014. HOG Feature Extractor Hardware Accelerator for Real-Time Pedestrian Detection. In *2014 17th Euromicro Conference on Digital System Design*. 543–550.

[11] C. Kyrkou and T. Theocharides. 2012. A Parallel Hardware Architecture for Real-Time Object Detection with Support Vector Machines. *IEEE Trans. Comput.* 61, 6 (June 2012), 831–842.

[12] S. Lee, H. Son, J. C. Choi, and K. Min. 2012. HOG feature extractor circuit for real-time human and vehicle detection. In *TENCON 2012*. 1–5.

[13] Seung Eun Lee, Kyungwon Min, and Taeweon Suh. 2013. Accelerating Histograms of Oriented Gradients descriptor extraction for pedestrian recognition. *Computers and Electrical Engineering* 39, 4 (2013), 1043 – 1048.

[14] G. Mishra, Y. L. Aung, M. Wu, S. K. Lam, and T. Srikanthan. 2013. Real-Time Image Resizing Hardware Accelerator for Object Detection Algorithms. In *2013 International Symposium on Electronic System Design*. 98–102.

[15] K. Mizuno, Y. Terachi, K. Takagi, S. Izumi, H. Kawaguchi, and M. Yoshimoto. 2012. Architectural Study of HOG Feature Extraction Processor for Real-Time Object Detection. In *2012 IEEE Workshop on Signal Processing Systems*. 197–202.

[16] M. Pedersoli, J. Gonzalez, X. Hu, and X. Roca. 2014. Toward Real-Time Pedestrian Detection Based on a Deformable Template Model. *IEEE Transactions on Intelligent Transportation Systems* 15, 1 (Feb 2014), 355–364.

[17] S. Sivaraman and M. M. Trivedi. 2013. Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis. *IEEE Transactions on Intelligent Transportation Systems* 14, 4 (2013), 1773–1795.

[18] Vladimir Vapnik, Steven E. Golowich, and Alex Smola. 1996. Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing. In *Advances in Neural Information Processing Systems 9*. MIT Press, 281–287.

[19] S. Wang, L. Duan, C. Zhang, L. Chen, G. Cheng, and J. Yu. 2016. Fast pedestrian detection based on object proposals and HOG. In *2016 International Joint Conference on Neural Networks (IJCNN)*. 3972–3977.