Design of embedded architecture for pedestrian detection in image and video

Taoufik Salem Saidani1,2 and Yahia Fahem Said2,

1Department of Computer Sciences, Faculty of Computing & Information Technology,
Northern Border University, Rafha, Saudi Arabia
2Laboratory of Electronics and Microelectronics (EμE), Faculty of Sciences of Monastir,
University of Monastir, TUNISIA

Summary
Today, pedestrian detection by real-time embedded systems remains a major challenge due to a number of factors. The task of detecting pedestrians in a road scene requires enormous time and resources. In this paper, a hardware architecture for pedestrian detection system is proposed. The system consists of a HOG descriptor extractor and an SVM classifier. The design is carried out using Xilinx's design tools: Vivado IPI, Vivado HLS and SDK for Hardware-Software Co-Design.
The performance analysis of the implementation shows a significant acceleration in the classification process with a reduction of the energy consumption and logical resources required. As a result, with the tools chosen, the proposed architecture has the capability to support a real-time pedestrian detection system for HD video at 180 frames per second.

Key words:
Pedestrian detection; HOG-SVM; Embedded architecture; Zynq APSoC; Real-time processing

1. Introduction

Human detection is a field of computer vision consisting of detecting humans in a digital image. It is a particular case of object detection, where we seek to detect the presence and precise location in image of one or more persons, usually in a posture close to that of the standing situation or a walk. It is also referred as pedestrian detection, because of the importance of video surveillance applications and for vehicle vision systems.

The detection of persons is a rather difficult subject, because of the wide variety of appearance of people, the articulation of the human body and occultation phenomena. In this sense, several studies have shown that detection must be carried out under difficult conditions and in an unconstrained environment, using low-quality imaging equipment: low-resolution CCTV cameras, embedded cameras on vehicle, etc. The problem is therefore to find a representation of a human being that is both generic enough to encompass all types of situations, and sufficiently discriminating to represent only humans. For this purpose, an intermediate representation is generally used, based on the computation of descriptors which are a set of scalar numbers generated to describe an object or a form. Characteristics of areas containing people are generally used by a supervised learning method to determine a person model. This model is then used to classify an area of the image as a person or not, from a feature vector (descriptor) calculated on this area. The process flow of supervised detection is presented in Figure 1.

Feature extraction is a major step in the detection process. The goal is to find the most relevant information to represent pedestrians. Numerous algorithms have been designed to provide more accurate and robust detection.

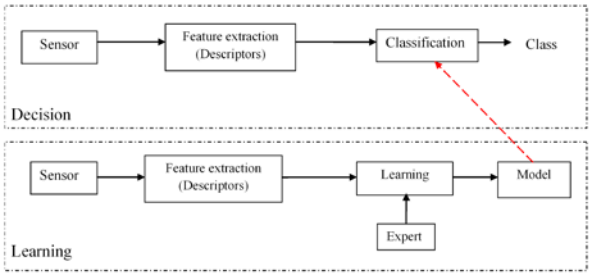

Fig. 1 Process flow of supervised detection

The SIFT (Scale invariant Feature Transformation) descriptor proposed by Lowe [1] captures the local scale and dominant orientation present in rectangular areas of interest by a voting method in a histogram of orientations. The histograms are weighted by the gradient norm and the descriptors thus obtained are invariant by scaling and rotation.

SURF (Speeded Up Robust Features) is a robust descriptor, presented by Herbert Bay et al. [2]. It is inspired by the SIFT descriptor. This descriptor is several times faster and more robust against different image transformations.

Dalal and Triggs [3] also work with histograms of gradient orientation. They offer a complete study on various parameters of their descriptor and provide a data base for the training and testing of pedestrian detection systems. This descriptor is computed in a dense manner on a uniformly divided grid and a process of normalization of the cells with multiple overlays brings robustness to the variations of luminosity. The histogram of oriented

Manuscript received December 5, 2017
Manuscript revised December 20, 2017

gradients (HOG) is one of the most successful algorithms in its class. It has been widely used for the detection of humans [9, 10].

The Local Binary Patterns (LBP) [4] operator is one of the most successful texture descriptors and has been widely used in various applications. The idea of this operator is to assign each pixel a code depending on the gray levels of its neighborhoods. The gray level of the central pixel (ic) of coordinates (xc, yc) is compared to that of its neighbors (in). This descriptor is robust against monotonous gray scale changes caused, and against lighting variations. Another important property that is its simplicity of calculation, which allows analyzing images in difficult settings in real time.

Several studies have shown that the simultaneous or alternating use of several types of descriptors allows a significant improvement of the results [5]. For example, the combined use of HOG and LBP provides better results than independent use because HOG works poorly when the background is cluttered with noisy edges. Local binary patterns are complementary in this aspect. It can filter noises using the concept of uniform pattern.

Characteristic analysis determines the class of data membership. It is thus a question of classifying the data. Numerous classification methods exist, each answering very specific problems. These will be the supervised learning classification methods, in order to realize the final application which is pedestrian detection in real time.

Support Vector Machine (SVM) is a two-class classification method [6] that attempts to separate positive and negative examples in all samples. The method then looks for the hyperplane that separates the positive examples from the negative ones, ensuring that the margin between the nearest positive and negative is maximal. This ensures a generalization of the principle because new examples may not be too similar to those used to find the hyperplane but be located on one side or the other of the border.

The Relevance Vector Machine RVM method was developed by Tipping [7]. It is a method that can also deal with regression problems. It uses the classical linear model of SVM kernel machines, but uses a Bayesian formulation to determine the parameters and select the relevant examples that will make the final discriminant model possible.

Boosting [8] is a method that combines numerous algorithms that rely on sets of binary classifiers: boosting optimizes their performance. The principle comes from the combination of classifiers (also called hypotheses). By successive iterations, the knowledge of a weak classifier is added to the final classifier (strong classifier). The classifier provided is weighted by the quality of its classification: the better it ranks, the more important it will be. Misclassified examples are boosted so that they become more important to the weak learner in the next round, so that he or she can make up for the lack.

The pedestrian detection system needs acceleration to enable real-time adaptive processing. Hardware acceleration has the potential to speedup these algorithms, making real-time processing for many image and video processing. The Hardware acceleration can be achieved using field programmable gate arrays (FPGA) or Graphic Processor Unit (GPU), which are devices consisting of reconfigurable hardware, allowing their function to be customized for a specific application.

For intensive computing, FPGAs have very large logical resources (multipliers, accumulators). In addition, they offer highly flexible architectures, they can easily divide the video source to independently feed the display (video output) or different additive blocks for subsequent video processing. Also, an FPGA can perform different processing with independent clocks without the need for additional resources for time multiplexing, unlike classical CPU or GPU processors.

Thanks to advanced semiconductor technologies, modern FPGA-SoC (Field Programmable Gate Arrays System on Chip) generations are powerful enough to support real-time image processing because of their high logical density, generic architecture and their memory on chip.

Today, faced with the integration density of FPGAs and the progressive demand for logical resources of advanced applications, it is very difficult or impossible to design IPs for embedded vision with the traditional hardware description language (HDL). Indeed, an implementation of an intensive image processing on FPGA requires a very important development time and generally leads to problems of reliability in the design. As a result, many efforts have been made to cope with this huge amount of resources through the integration of tools that offer a design flow at a higher level of abstraction than traditional HDL.

Vivado High-Level Synthesis Tool speeds up the creation of IPs by allowing C, C ++ and SystemC specifications to be directly targeted in all Xilinx All Programmable SoC FPGAs without having to create the RTL manually. It offers an opportunity to go faster to IP creation while exploiting its properties.

With the introduction of reconfigurable platforms such as AP SoC and the advent of new high-level tools for configuring them, FPGA-SoC image processing has emerged as a practical solution for most computer vision problems and image processing. In this context, we were interested in the design and implementation of an embedded video processing architecture. This research aims to propose an embedded architecture of a pedestrian detection algorithm on a hardware/software co-design platform suitable for use as an embedded system.

The remainder of this paper is organized as follows. Related works on embedded architecture for pedestrian

detection are reviewed in Section 2. Section 3 describes the hardware implementation design of the proposed pedestrian detector. Finally, Section 4 and 5 present the results and conclusions respectively.

2. Related Works

The proven performance of human detection algorithms has motivated many groups of researchers to offer equivalent hardware architectures for integration into real-world applications such as driver assistance systems. Thus, to speed up the extraction of features and reduce computational complexity, several methods have been proposed. This section reviews related works on embedded architecture for pedestrian detection.

In recent years, the frequency of central processors (CPU) has been capped, for thermal reasons, at about 4 GHz. This limitation has been circumvented by a change in the internal architectures of the processors. The performance gain no longer passes by an increase in frequencies, but by the multiplication of the calculation units integrated into the processors.

Due to runtime problems, the searches and implementations of the algorithms are oriented to be implemented on parallel architectures and more precisely on the Graphic Processing Unit (GPU) because of its high efficiency and its ability to accelerate the processing time of different applications.

GPU computing can parallelize tasks and provide maximum performance in many applications. The GPU accelerates the heavier parts of the code, the rest of the application remains assigned to the CPU. For this reason, the researchers have oriented their work on GPU to ensure a gain in execution time and a great acceleration of their systems of people detection.

Parisacariu et al. [11] proposed a pedestrian detection system based on the histogram of oriented gradients descriptor (HOG) and the Support Vector Machine (SVM) classifier. The entire algorithm was implemented on an NVIDIA GPU using CUDA as the programming language. They have achieved encouraging results and accelerations that can go up to 67 times compared to sequential CPU processing.

The Adaboost cascade algorithm is widely used for detecting people in images and videos. Cai et al. [12] presented an implementation of the Adaboost cascade algorithm in combination with the Haar descriptor. The implementation of their detection system was based on NVIDIA CUDA technology. In this work, Cai et al. have completed accelerations up to 6 times compared to a CPU implementation.

In [13], Choi et al. process an approach for detecting and tracking people in infrared images and video on GPUs. The detection phase is performed by following two algorithms based on two combinations: the first consists of the Adaboost classifier in combination with the Haar descriptor. The second is based on the SVM classifier in combination with the HOG descriptor.

Fassold et al. [14] proposed an efficient GPU implementation of the Scale-Invariant Feature Transform (SIFT) algorithm using CUDA for programming GPUs. Knag et al. [15] presented a parallelization and optimization of SIFT and SURF algorithms on GPU using OpenCL [16] and OpenGL [17] as programming languages for GPUs.

In [18], Campany et al. used a new person detection system designed primarily for driving assistance, based on the HOG, LBP and HOG descriptors and LBP together and SVM as a classifier. To evaluate the performance of the three person detectors, the authors implemented their algorithms on GPU for video sequences containing images of 1242 x 375 pixels. They used CUDA as the programming language for their implementations. The implementation of these people detectors was performed on two platforms that are: Tegra x1 and GTX 960.

Zhang et al. propose in [19] a system for detecting and re-identifying people in images and videos on GPU. This system is based on the SURF descriptor and the Convective Neural Network classifier (R-CNN) [20].

Real-time image-based human detection plays a key role in a wide range of application domains. We distinguish a lifting of the importance of the implementation of the people detection algorithms to ensure a real-time detection. FPGA implementations are a good choice and a promising approach for robust, real-time detection. In the following we present some implementations of person detection algorithms on FPGA.

Hahnle et al. propose in [21] an FPGA-based real-time implementation of their HOG based people detection system in combination with SVM. Reliable and rapid detection of pedestrians in mobile systems is an important and growing area of research. Increased requirements for road safety make pedestrian detection necessary on mobile systems. In this context Piao et al. [22] implemented their human detection algorithm for mobile systems. Their detection system was based on a system on chip (SoC) on Linux. This system combines the advantages of parallel computing by using a hardware implementation on FPGA.

Another implementation of the HOG descriptor in combination with the SVM classifier has been proposed by Li et al. [23]. They proposed a new approach based on binarization (digitization) of data to simplify the calculation process. This approach is specifically designed to minimize the calculation and subsequently minimize the requested memory space. A partial dynamic reconfiguration based on a Xilinx FPGA has been developed in this work.

Tasson et al. [24] have implemented one of the most effective models for the person detection, it is the DPM

(Deformable Part Model) on FPGA. This allowed them to complete considerable detection accelerations as well as robustness in detection. Their system has undergone modifications to process deformed images of people to ensure a very robust detection.

A person detection system based on the region covariance descriptor and the SVM classifier is presented in [25]. The authors present a hardware implementation based on a Xilinx vitrtex-4 FPGA. The presented algorithm is based on the concept of the sliding window, this allows detecting and characterizing the pedestrian in a robust way. All algorithms based on the concept of the sliding window suffer from problem at runtime level. For this reason the authors have tried to speed up the detection process by adopting an implementation on FPGA.

Said et al. propose in [26] an effective implementation of a pedestrian detection system based on the SCoM-SVM descriptor following a concept based on sliding window method. The implementation of this detection system was based on a Xilinx Spartan 3A-DSP platform. The results obtained show a high processing performance and a high detection accuracy.

3. Hardware Implementation Design of the Proposed Pedestrian Detector

3.1 Pedestrian Detector Algorithm

The proposed detection system is divided into two parts: a HOG descriptor extractor and an SVM classification system operating with a sliding windowing technique. The first part presented in Figure 2 transforms the input image into a set of descriptors. Each HOG descriptor will be associated here with a certain zone of the image, called block. In order to detect all the objects present in the image, the scene is divided into a collection of detection windows on which the probabilities of presence of the searched object will be evaluated. The input vector associated with each detection window is then composed of the aggregation of all the HOG descriptors from the blocks belonging to the window (see Figure 3). The SVM calculates the scalar product between the input vector and the model from the supervised learning in order to make a decision about the presence of the object sought in the window.

At the highest level, the detection system studied is described in the form of a data flow graph in Figure 4. The incoming pixel stream is processed in the descriptor extraction module which generates a stream of HOG descriptors. In order to determine all occurrences of the target object in the image, the detection window is moved in the image and for each position, the system evaluates the presence of the targeted object. The images enter the graph from the left as a stream of pixels and the system produces a stream of binary images where each bit corresponds to the response for the associated detection window.

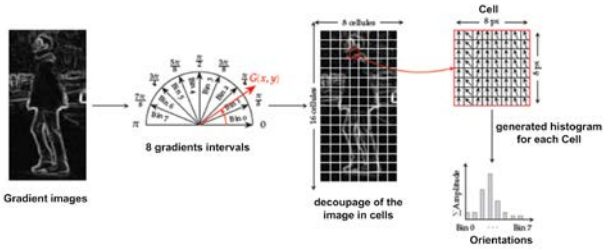

Fig. 2 Calculation of the histogram of gradient orientations

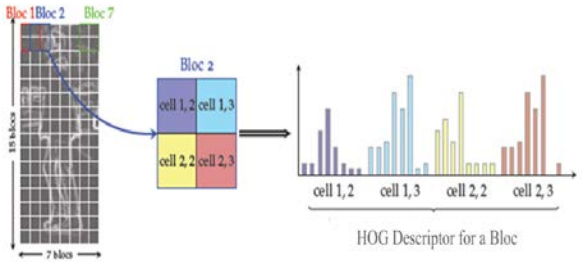

Fig. 3 Grouping cells into blocks

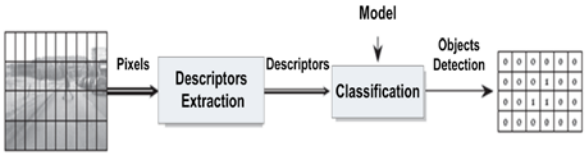

Fig. 4 Overview of the data flow object detection system

The reformulation of the HOG descriptor calculation step in accordance with the data flow model is presented in Figure 5. The images enter as structured streams of pixels and the outputs are output as structured flows of descriptors, where each component of the descriptor is carried by a parallel flow.

The graph of Figure 5 by contrast highlights the parallelism of the algorithm. The chosen approach takes advantage of the fact that each entry of the histogram constituting the descriptor can be calculated independently of the others (in parallel). To do this, we replicate a chain of specific actors on each entry. This chain of actors corresponds to three stages, dedicated respectively to the calculation of the gradient (actors conv and argmax), to the calculation of the histogram (actors hsum and vsum) and to the normalization of the descriptor (actors block, norm_factor and div).

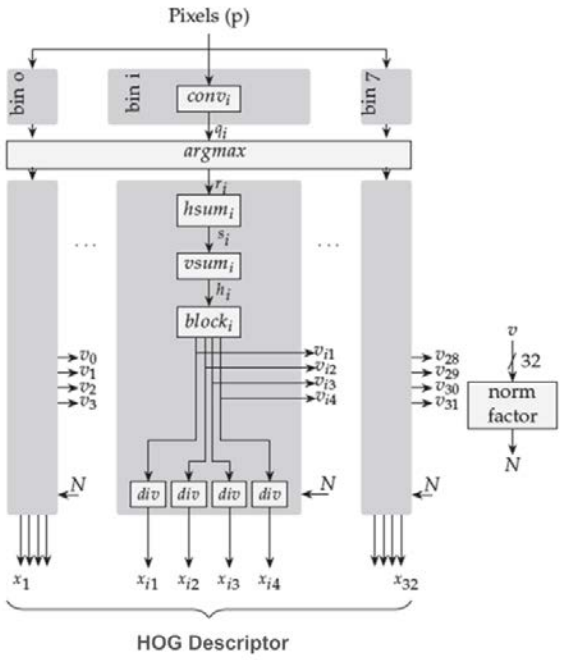

Fig. 5 Data Flow Graph of HOG Descriptor Extraction

The determination of the full descriptor associated with a window is done by aggregating all the descriptors of the blocks included in this window. In order to evaluate the result of the detection on a window, it is necessary to calculate the scalar product of the complete descriptor associated with this window with the model resulting from the learning phase.

A first stream data formulation of the system that will be the basic building block of a complete data flow system managing overlaps, the management of recoveries is done by simply putting in parallel of 105 basic bricks (nx x ny in our case) . The Data flow graph describing the operation of this basic brick is given in Figure 6.

3.2 Method of implementation

In this work, we used the heterogeneous platform ZC702. It is a platform based on the Zynq-7000 Programmable System-On-Chip (SoC) of technology 28 nm. The latter merges the benefits of processor and programmable logic through the integration of a dual-core ARM 9 processor and an FPGA into a single chip. Such an architecture offers the power to work in parallel mode of an FPGA, with a guarantee of flexibility and ease of implementation of the software applications of the processor.

A Vita-2000 camera is coupled with the ZC702 platform. It is an 'Ultra eXtended Graphics Array' CMOS image sensor (WUXGA) configurable in HD (1920 x 1080) or 1600 x 1200 formats. To ensure connectivity between the image sensor and the embedded platform, an FPGA Mezzanine Card FMC module is used. It has two types of interfaces: an LCD Coaxial Embedded Display Interface (LCEDI) and two HDMI input / output interfaces

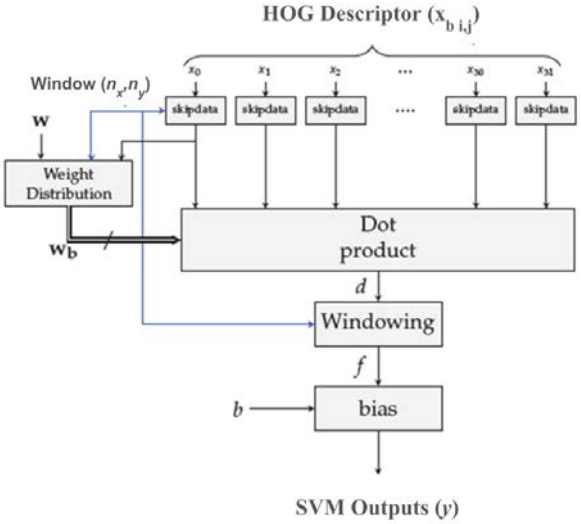

Fig. 6 Data flow graph of classification system

In parallel with the new generation of Zynq-7000-based platforms, Xilinx provides a suite of design tools to establish a direct link between the hardware and software part of the system. Indeed, the mixed architecture of Zynq requires a tool that uses the various programming languages (HDL, C and C ++) for the joint design of the software and hardware level. This Co-design approach makes it possible to simultaneously develop part of the application with a hardware description language and another part with software using the high level language.

The design flow for Zynq SoC is presented in Figure 7. The first step is to define the desired behaviors of the system, that is, to create an appropriate specification from a set of requirements. This is represented as the starting point at the top of the diagram and is the basis of the system design developed later.

The Zynq architecture combines an ARM processor (for the software elements of the designed system) with an FPGA (mainly for the hardware elements of the system). A key element of the next system design step is therefore to properly partition the intended functionality between software and hardware and to define the interfaces between the two sections. Of course, it is possible that this partitioning is later adjusted by the designers. After partitioning the system, the development of software and hardware can progress in parallel, to a large extent. With respect to hardware development, it is necessary to identify the functional blocks necessary to achieve the design and then to assemble them through a combination of design reuse and Intellectual Property development (IPs) of new technologies and technologies to create appropriate connections between the blocks. Similarly, the

software aspect of the project can be achieved by developing custom code or reusing pre-existing software. Verification of software and hardware will be required, which is an integral and important part of the process. The hardware and software elements of the systems must be integrated, according to the interfaces defined at the stage of the specification, as well as "complete system" tests undertaken.

In order to solve the programmability problems of FPGAs, a great number of researches have been undertaken on automatic hardware synthesis systems from higher level languages. These systems are commonly known as High Level Synthesis (HLS). Most tools, industrial or academic, offer direct conversions from high-level language (C / C ++) to HDL code.

Xilinx provide a number of tools which enable the creation of custom IP blocks for use in the embedded system designs. Vivado HLS is a tool provided by Xilinx, as part of the Vivado Design Suite, which is capable of converting C-based designs into RTL design files (VHDL/Verilog or SystemC) for implementation of Xilinx All Programmable devices. An overview of the Vivado HLS design flow is provided in Figure 8.

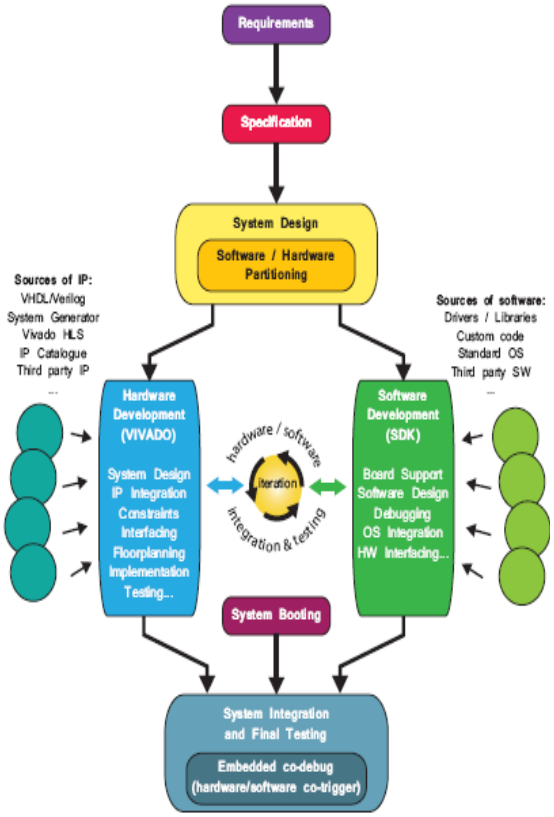

Fig. 7 The design flow for Zynq SoC

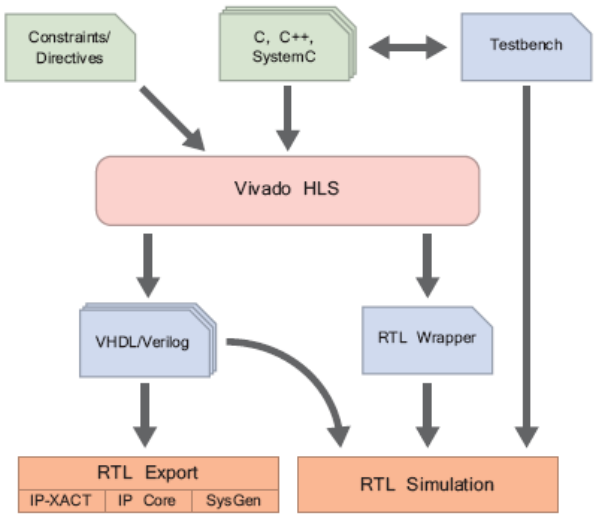

Fig. 8 An overview of the Vivado HLS design flow

The design of an IP block also requires source codes, the OpenCV library offers these codes in C or C ++. These source codes are the essential and the basis of our design, Figure 9 presents a screenshot of the source code of the HOG-SVM algorithm that will be synthesized by Vivado HLS in order to implement it as a hardware accelerator.

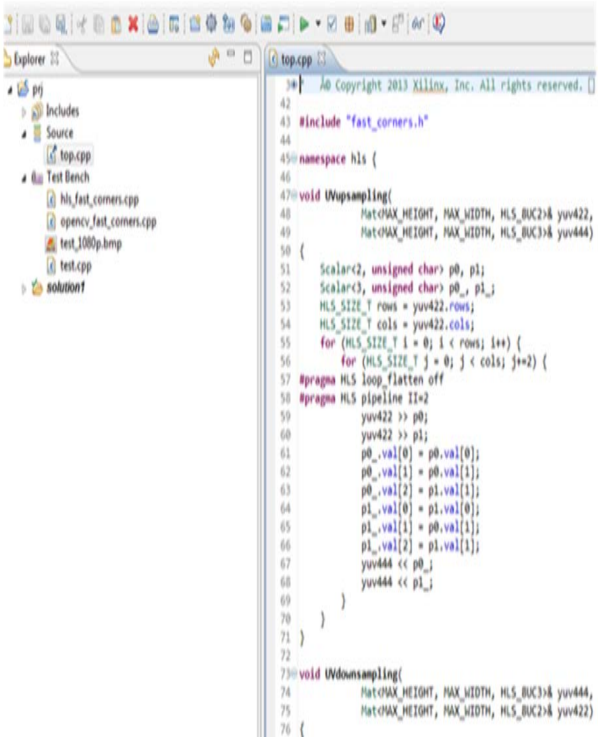

Fig. 9 HOG-SVM source codes

The proposed HOG-SVM hardware architecture was created, packaged, and added to the Xilinx IP library as a customized IP using the Vivado IP Integrator tool and then implemented in the Xilinx part (PL) and connected to the ARM processor via the AXI busses. An overview of our pedestrian detection system implemented in Zynq through the Vivado environment is presented in Figure 10. The system includes a processor and its associated peripherals to manage the software application, synchronization, conversion and the adaptation of the different modules, HDMI input for video acquisition, HDMI output for display and the HOG-SVM IP Core.

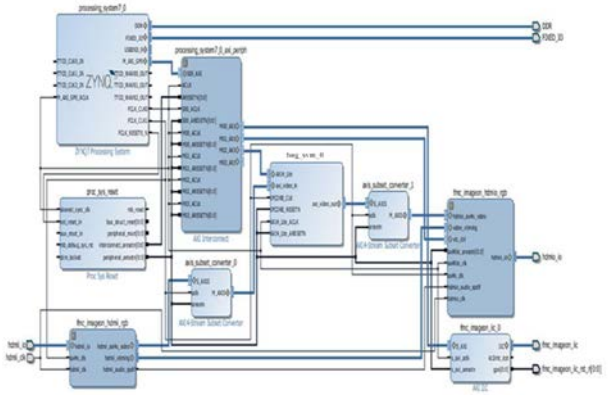

Fig. 10 Overview of the pedestrian detection system in the Vivado environment

The majority of the IPs used in the project having a flow of pixels for input and output, we have thus connected these IPs to each other by the standard AXI-Stream interface which groups the pixel value, its X and Y coordinates and a signal "Enable". This allows a seamless connection between its IPs and a certain modularity of the overall architecture of the system.

Once the system is set up in the Vivado environment, we must then transfer the Hardware specifications to SDK so that we can attach it to the dedicated C language (.h and .c) files to drive the developed IPs and I/O devices of the system.

4. Implementation Results and Evaluation

The implementation of the proposed pedestrian detection system was made on the Xilinx ZC702 platform. Xilinx Vivado 2015.2 was used for logical synthesis and placement and routing. The logical resources used for the IP HOG-SVM and for the entire detection system are shown in Table 1 and Table 2, respectively. The system uses only 35% of the Flip Flops registers, 21% of the DSP blocks. , 52% of LUTs and 68% of BRAMs. The resource utilization summary shows that, with the proposed pedestrian detection architecture, the remaining FPGA resources are sufficient for the addition of other parallel processing on the same platform, such as adding a step extra for generating multi-scale samples from a single detection window, which can increase the accuracy of the system.

The using rate of material resources for the implementation of our pedestrian detection system on the PL of the Zynq ZC702 platform reflects the effectiveness of the proposed methodology.

Table 1: The implementation results on the Zc702 platform

Resource Type	Availble	Used	%
BRAM	140	94	68
DSP48E1	220	46	21
Flip Flops	106400	63608	35
LUTs	53200	27780	52
Memory LUT	17400	4500	26
Maximum Frequency	444 MHz		

Table 2: Quantity of hardware resources used by the IP HOG-SVM

Resource Type	Availble	Used	%
BRAM	140	60	43
DSP48E1	220	39	18
Flip Flops	106400	29648	28
LUTs	53200	16280	31
Memory LUT	17400	3400	20

A comparison of the results of our implementation with the proposal of Kelly et al. [30] is presented in Table 3. As shown in this table, the design of Kelly et al. [30] requires 41% of DSP blocks and 21% of LUTs with a clock frequency of 404 MHz. On the other hand, our resulting architecture required about 21% of the DSP blocks and 52% of the LUTs with a working frequency up to 444 MHz.

Table 3: Performance comparison

Architecture	Our's		[30]	
Resource Type	**Used**	**%**	**Used**	**%**
BRAM	94 (of 140)	68	24 (of 140)	18
DSP48E1	46 (of 220)	21	90 (of 220)	41
LUTs	27780 (of 53200)	52	10694 (of 53200)	21
Maximum Frequency	444 MHz		404 MHz	

The maximum frequency of the integrated PS part on the Zynq ZC702 platform is approximately 866 MHz, while the maximum frequency of the PL part depends on the design to be implemented. In our case, the maximum propagation delay has been estimated at about 2.25 ns, resulting in a maximum frequency of 444 MHz that supports HD resolution at 180 frames per second. However, in our implementation, hardware IPs operate at the minimum frequency of 148 MHz (60 fps HD video acquisition rate via the HDMI interface) which is sufficient to meet the real-time constraints of the

pedestrians detection application. Clearly, the use of one cell scan and the simultaneous partial SVM classification contribute significantly to reducing the number of cycles required. Indeed, when the stream of pixels reaches the level of a window, only one additional cycle is necessary to calculate the HOG characteristic vector and another for the classification.

Another important benefit of maintaining a low operating frequency is the reduction of the power consumption of the PL part. For system evaluation, the total energy consumption is analyzed statically and dynamically. As for static consumption, it is due to the leakage current of the transistors, so it is directly related to the total number of logical resources used in the design. As resources increase, the static energy consumption becomes higher. While dynamic power is caused by all types of signal transition, such as clocks, logical resources, memories (BRAM) and I/O. As a result, higher clock rates lead to higher dissipation. The total energy consumption (static and dynamic) of the proposed HOG-SVM architecture is 1.532W. This value is 2.3 times smaller than the consumption achieved in [30], although the authors only implemented HOG (without SVM) in the PL part of the same Zynq device.

In this work, an offline learning step of the linear SVM classifier was performed with the INRIA [28] and MIT [29] databases. The bias (b) and weight coefficients (w) defined in equation (1) are the results of the system after learning. These two parameters will be stored in an internal memory of the FPGA for use in the online decision phase. Indeed, the decision is translated in practice by the scalar product calculation of the vector HOG and weight coefficients SVM by using the following equation:

$$y(x) = w^T . x + b \quad (1)$$

Where w^T and b are determined during offline learning, x is the calculated HOG vector for a new unknown sample.

The hardware implementation is tested on the database Daimler Pedestrian Benchmark Dataset [27]. From this base, we extracted a sequence of images where a pedestrian crosses the roadway. This selection is explained by the need to test the effect of the parameters and steps of the algorithm under relatively controlled conditions. Figure 11 illustrates an example of detection on an image of the chosen sequence. This result validates the last step of the proposed methodology, namely the transition from a VHDL RTL simulation to an implementation on a real platform.

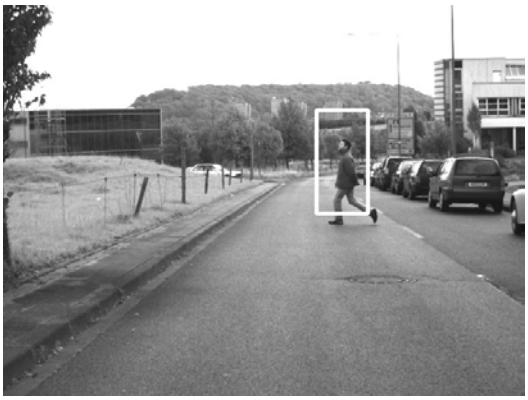

Fig. 11 Example of detection of hardware implementation on Daimler base

5. Conclusion

Taking into account the requirements of embedded systems of image processing in general, and systems of detection and recognition of people in particular, it is essential to share and synchronize the different parts of necessary treatment on heterogeneous resources such as a processor and an FPGA. Indeed, the processor is adapted to perform some software processing, control access to the memories and connect to the outside world, while the programmable logic part is well suited for intensive data processing such as description and classification of images. In this context, a complete architecture of the HOG-SVM couple allowing a considerable reduction of the logical resources and an acceleration of the computation time will be proposed and implemented on a FPGA of the family Zynq-7000 AP SoC in order to complete the goal of pedestrian's detection in real time.

Deployment on a Xilinx Zynq ZC702 platform has shown that a High Level Synthesis HLS tool enables efficient implementation of the proposed detection algorithm on an FPGA target. The generated code, fully automatic, allows operation at 180 frames per second with HD resolution.

The tests performed to quantify the quality of the pedestrian detection system were applied to a sample of the Daimler base. This choice was made mainly in order to control a certain number of parameters during the tests. However, a statistical evaluation of the performances on the whole database is to be considered in the future in order to obtain more precise results on the reliability of our system.

Acknowledgments

The authors wish to acknowledge the approval and the support of this research study by the grant N° CIT-2017-

1-7-F-7037 from the Deanship of the Scientific Research in Northern Border University, Arar, KSA.

References

[1] Lowe, D.G.: 'Distinctive image features from scale-invariant keypoints', Int. Journal of Computer Vision , 2004, 60, (2), pp. 91-110.

[2] H. Bay, T. Tuytelaars and L. Van Gool ."SURF: Speeded Up Robust Features", Proceedings of the 9th European Conference on Computer Vision, Springer LNCS volume 3951, part 1, pp 404--417, 2006.

[3] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection ''.In CVPR, 2005, p: 886-893.

[4] T. Ojala, M. Pietikäinen and T. Mäenpää. "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns". IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, p: 971–987, 14, 66, July 2002.

[5] X. Wang Tony X. H. Sh. Yan "An HOG-LBP Human Detector with Partial Occlusion Handling".

[6] C. Cortes and V. Vapnik. "Support-vector networks''. Machine learning, 20 (3), pp: 273-297, 1995.

[7] M. E. Tipping. "Sparse bayesian learning and the relevance vector machine". Journal of Machine Learning Research, , pp:211–244, 6 May 2001.

[8] Robert, E.S., Yoram, S.: 'Improved boosting algorithms using confidence-rated predictions', Machine Learning, 1999, 37, (3), pp. 297-336.

[9] Wu, B., Nevatia, R.: 'Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors'. ICCV, Vol. 1, 90-97. (2007)

[10] Dollár, P., Wojek, C., Schiele, B., Perona, P.: 'Pedestrian detection: an evaluation of the state of the art'. IEEE Trans. Pattern Anal. Mach. Intell., Vol. 34, No. 4, 743-761. (2012).

[11] Prisacariu, V., & Reid, I. 'Fast HOG: a real-time GPU implementation of HOG'. Department of Engineering Science, 2310(9), 2009.

[12] Cai, C. C., Gao, J., Minjie, B., Zhang, P., & Gao, H. 'Fast Pedestrian Detection with Adaboost Algorithm Using GPU'. International Journal of Database Theory and Application, 8(6), 125-132. (2015).

[13] Choi, B. J., Yoon, B. W., Song, J. K., & Park, J. (2015). 'Implementation of Pedestrian Detection and Tracking with GPU at Night-time'. 20(3), 421-429.

[14] Fassold, H., &Rosner, J. (2015, February). 'A real-time GPU implementation of the SIFT algorithm for large-scale video analysis tasks'. In Real-Time Image and Video Processing.

[15] Kang, S. H., Lee, S. J., & Park, I. K. (2014, January). 'Parallelization and optimization of feature detection algorithms on embedded GPU'. In International Workshop on Advanced Image Technology (Vol. 108, pp. 164-167).

[16] K. Group. Open Computing Language (OpenCL). http://www.khronos. org/opencl/.

[17] K. Group. OpenGL ES. http://www.khronos.org/opengl/.

[18] Campmany, V., Silva, S., Espinosa, A., Moure, J. C., Vázquez, D., &López, A. M. (2016). 'GPU-based pedestrian detection for autonomous driving'. Procedia Computer Science, 80, pp: 2377-2381.

[19] Zheng, Y., Chen, Z. J., Velipasalar, S., & Tang, J. (2016, September). 'Person Detection and Re-identification Across Multiple Images and Videos Obtained via Crowd sourcing'. In Proceedings of the 10th International Conference on Distributed Smart Camera (pp. 178-183). ACM.

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik. R'ich feature hierarchies for accurate object detection and semantic segmentation'. In Proceedings of the IEEE Conference onComputer Vision and Pattern Recognition (CVPR), 2014.

[21] Hahnle, M., Saxen, F., Hisung, M., Brunsmann, U., & Doll, K. (2013). 'FPGA-based real-time pedestrian detection on high-resolution images'. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 629-635).

[22] Piao, S., Kiekbusch, L., Schmidt, D., Berns, K., Hering, D., Wirtz, S., Weiland, J. (2016, March). 'Real-time multi-platform pedestrian detection in a heavy duty driver assistance system'. In Proc. Int. Commercial Veh. Technol. Symp. (pp. 61-70).

[23] Li, Y., Gai, K., Qiu, M., Dai, W., & Liu, M. (2017). 'Adaptive human detection approach using FPGA-based parallel architecture in reconfigurable hardware'. Concurrency and Computation: Practice and Experience, 29(14).

[24] Tasson, D., Montagnini, A., Marzotto, R., Farenzena, M., & Cristani, M. (2015). 'FPGA-based pedestrian detection under strong distortions'. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 66-71).

[25] Martelli, S., Tosato, D., Cristani, M., & Murino, V. (2011, August). 'FPGA-based pedestrian detection using array of covariance features'. In Distributed Smart Cameras (ICDSC), 2011 Fifth ACM/IEEE International Conference on (pp. 1-6). IEEE.

[26] Said, Y., & Atri, M. (2016). 'Efficient and high-performance pedestrian detector implementation for intelligent vehicles'. IET Intelligent Transport Systems, 10(6), pp: 438-444.

[27] Markus Enzweiler and Dariu M Gavrila. 'Monocular pedestrian detection: Survey and experiments'. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 31(12), pp: 2179–2195, 2009.

[28] INRIA Person Dataset. http://pascal.inrialpes.fr/data/human/

[29] CBCL pedestrian database. http://cbcl.mit.edu/software-datasets/PedestrianData.html

[30] Kelly, C., Siddiqui, F. M., Bardak, B., and Woods, R.: 'Histogram of oriented gradients front end processing: An FPGA based processor approach'. IEEE Workshop on Signal Processing Systems (SPS), pp: 1-6, 2014.

Taoufik Salem Saidani received the M.S. and Ph.D degrees in Electronics from Faculty of Science of Monastir, Tunisia in 2007 and 2014, respectively. His major research interests include VLSI and embedded system in video and image compression. His current research interests include digital signal processing and hardware-software co-design for rapid prototyping in telecommunications.

Yahia Fahem Said received the M.S and Ph.D Degrees in Electronics from Faculty of Science of Monastir, Tunisia in 2010 and 2016, respectively. His research interests are focused on the study and development of FPGA-based real time implementation of video and image processing applications. Specifically, His areas of interest include: Image and Video Processing, SoC Design, Embedded Vision, Pattern Recognition, Artificial Intelligence and Deep Learning.