Hardware Acceleration of SVM-Based Classifier for Melanoma Images

Shereen Afifi[1(✉)], Hamid GholamHosseini[1], and Roopak Sinha[2]

[1] Department of Electrical and Electronics Engineering,
School of Engineering, Auckland University of Technology,
Private Bag 92006, Auckland 1142, New Zealand
{safifi, hgholamh}@aut.ac.nz

[2] School of Computer and Mathematical Sciences,
Auckland University of Technology, Auckland 1142, New Zealand
rsinha@aut.ac.nz

Abstract. Melanoma is the most aggressive form of skin cancer which is responsible for the majority of skin cancer related deaths. Recently, image-based Computer Aided Diagnosis (CAD) systems are being increasingly used to help skin cancer specialists in detecting melanoma lesions early, and consequently reduce mortality rates. In this paper, we implement the most compute-intensive classification stage in the CAD onto FPGA, aiming to achieve acceleration of the system for deploying as an embedded device. A hardware/software co-design approach was proposed for implementing the Support Vector Machine (SVM) classifier for classifying melanoma images online in real-time. The hybrid Zynq platform was used for implementing the proposed architecture of the SVM classifier designed using the High Level Synthesis design methodology. The implemented SVM classification system on Zynq demonstrated high performance with low resources utilization and power consumption, meeting several embedded systems constraints.

Keywords: SVM · CAD · Melanoma · FPGA · Hardware implementation

1 Introduction

Computer Aided Diagnosis (CAD) systems have been widely used in practical clinical settings to support detecting of a variety of cancers. Melanoma is considered the most dangerous form of skin cancer, in which early diagnosis could help in dramatically decreasing morbidity and mortality [1]. Therefore, CAD of medical images should be efficiently exploited for improving early detection of melanoma lesion. Developing such systems for this kind of applications is extremely computationally demanding with high inherent parallelism. Systems with parallel processors could be useful for accelerating such computations, but they are energy intensive and have prohibitively high costs. Special purpose hardware such as reconfigurable hardware is promising for speeding up computations, and provides High Performance Computing (HPC) at lower cost and lower power consumption [2].

F. Huang and A. Sugimoto (Eds.): PSIVT 2015 Workshops, LNCS 9555, pp. 235–245, 2016.
DOI: 10.1007/978-3-319-30285-0_19

Field-Programmable Gate Arrays (FPGAs) are powerful massively parallel processing reconfigurable devices which are used for achieving necessary performance of embedded systems with efficient utilization of hardware resources. FPGAs have recently shown very high performance for many applications in image processing [3]. Accordingly, FPGA is considered to be a suitable hardware platform with HPC for implementing image processing algorithms with inherent parallelism essential for developing CAD of medical images.

This work is based on our previous work and experimental results achieved for diagnosis of melanoma [4]. A dataset of benign and malignant melanoma images was used for our CAD system which was collected from available web resources. The typical structure of a CAD system includes image pre-processing, segmentation techniques, features extraction, and classification schemes. Some methods were employed for the pre-processing, segmentation, and feature extraction stages in our previous work. Regarding the classification stage, five different classifiers (k-Nearest Neighbors, Multi-Layer Perceptron, Naive Bayes, Random Forest and Support Vector Machine (SVM)) were tested and evaluated for their classification performance for the selected melanoma dataset. From the performance comparison of the five classifiers that was based on different feature selection sets, the SVM classifier demonstrated a better accuracy than others for classification and diagnosis of melanoma [4]. Therefore, the SVM classifier is considered in this work as being one of the best classifiers with high accuracy level of classifying medical images targeting melanoma detection. Also, the SVM model is a powerful machine learning tool that was used for efficient classification in various applications [5].

Regarding acceleration of the CAD system, the classification scheme is the most compute-intensive task among the diagnosis process. For that reason, this work proposes hardware acceleration for the SVM classifier by implementing its time-consuming computation part on FPGA as a coprocessor targeting deployment in an embedded environment.

A hardware/software co-design is proposed for running the compute intensive task included in the SVM algorithm on the FPGA. The system is implemented on the recent Xilinx Zynq device using the High-Level Synthesis (HLS) design methodology. Experimental results demonstrated minimal hardware resources utilization and low power dissipation which increases the potential of developing an embedded CAD system.

The main contribution of this paper is proposing a hardware/software co-design of the SVM classifier implemented on the hybrid Zynq platform using HLS, targeting melanoma detection.

This paper is organized in five sections. Related work of hardware implementations for the SVM classifier on FPGA is briefly introduced in Sect. 2. The proposed hardware accelerated SVM classification system is presented in Sect. 3, followed by experimental results and discussion presented in Sect. 4. Finally, Sect. 5 provides the conclusion and future work.

2 Related Work on FPGA-Based Implementation of SVM Classifier

There are several research works that aim to speed up SVM classification in hardware. Various techniques of hardware architectures and designs have been implemented on FPGAs achieving high level of parallelization as well as high performance computing. Many work exploit the FPGA-based parallel systolic array architecture in their implementations [6–10], resulting in good results of classification speedups that mostly outperformed implementations on GPPs/CPUs. In addition, other work employed the Dynamic Partially Reconfiguration (DPR) technique [7, 11], achieving higher flexibility and design space expansion besides gaining speedups. Some research work also recorded relatively good results of power consumption reduction [6] as a result of using the DPR technology.

Interestingly, many studies adopted the multiplier-less approach [9, 12–14], where expensive multipliers required for computations are replaced with conventional adders and/or shifters in order to decrease hardware complexity. Similarly, others [15, 16] utilized the hardware-friendly kernel function for simplifying the hardware design by using also the simple shift and add operations instead of resource consuming multiplications. As a result of multiplier-less implementation, significant reduction in hardware resources utilization was achieved. In addition, remarkable power consumption decrease was demonstrated [9, 12, 13, 17].

Moreover, the common pipelining technique was used in many previous hardware designs [11–13, 17–26], taking advantage of the parallel processing capabilities of the FPGA that led to throughput increase of the implemented classification process. Some researchers designed a pipeline stage for common and shared multipliers required for computations to decrease usage of duplicate multiplications [19, 20]. Some pipelined designs [21, 23] were based on exploiting the embedded Intellectual Property (IP) cores in the FPGA device for efficient resources utilization. Furthermore, a pipelined adder structure was used by various designs [11–13, 17, 18, 24].

To summarize, existing implementations in literature for SVM classification target improvement in accelerating the classification process. However, there is a trade-off between classification accuracy and high performance, and meeting embedded systems constraints of low level of area and power consumption. Due to some simplifications performed for the hardware implementation, some loss in accuracy rating occurs [12, 13, 17, 26, 27]. Moreover, most designs in existing literature were implemented on old versions of FPGAs and only a very limited number used recent ones [9, 28].

To the best of our knowledge no implementation for SVM classification phase exists in literature that applies a hardware/software co-design technique taking advantage of the hybrid architecture of the Zynq platform. In addition, all previous implementations were designed by using the traditional Hardware Description Language (HDL), which is time consuming and requires expert hardware designers. However, HLS design methodology has been lately recommended for simplifying hardware designs for coprocessors on FPGAs [29]. Moreover, no hardware implementation on FPGA exists in literature for the SVM model to classify melanoma clinical images. Consequently, this work proposes a hardware/software co-design method for implementing SVM classifier on Zynq device using HLS method targeting detection of melanoma with high performance and low cost.

3 The Hardware Accelerated SVM Classification System

3.1 Selected Platform and Design Tools

One of the most recent platforms; Zynq-7000 was selected for implementing the SVM classifier on hardware, complying with latest technology to reach an efficient embedded system. The Zynq-7000 All Programmable System on Chip (SoC) is one of the recent platforms provided by Xilinx [30]. It has a hybrid structure of combining an ARM Cortex-A9 dual-core as a Processing System (PS) with the hardware programmability of an FPGA as a Programmable Logic (PL) in a single SoC, which simplifies embedded systems realization. Accordingly, the Zynq-7000 SoC is considered to be a promising platform for creating hand-held medical imaging systems with high performance and low cost.

Additionally, Xilinx offers very efficient and powerful software tools; Vivado Design Suite that simplifies the embedded system design process on a single device, integrating a SoC with an FPGA [31]. Interestingly, Xilinx provides an efficient tool employing the HLS design methodology, which simplifies programing FPGA with high-level language than using the traditional HDL [29]. Thus, using the HLS method is recommended for significantly decreasing FPGA development time and effort [32]. Therefore, the Xilinx Vivado HLS tool was exploited to develop a hardware accelerator for speeding up the SVM classification process to be integrated within a single SoC and realized on the recent Zynq device.

3.2 The Employed SVM Classifier

One of the common SVM classifiers algorithms; SVM-light was used to implement a case study to classify given melanoma images. SVM-light is an available simple classifier implemented in C language which is used for classification in different domains [33]. The C source code of the SVM-light classification phase was adopted for the hardware implementation using the HLS method in order to reach hardware acceleration.

The classification code of a binary SVM-light model with linear kernel was studied for customization in order to reach a synthesizable code capable of realization on hardware. Thus for synthesizing, the dynamic memory allocation scheme that applies in the C code for flexibility was changed for static fixed sizes targeting our case study. As a result, the algorithm became dependent on the maximum number of selected features (vector size) which should be defined before implementation.

The most time and resource consuming part in the classification algorithm is the accumulation/summation of multiplication of vectors that exists in the main decision function (1) using linear kernel (dot product) [34].

$$\text{Class (s)} = \text{sign}\left(\sum \alpha_i y_i (x_i .\ s) + b\right) \quad (1)$$

To classify a new pattern known as *s* in (1), the dot product between *s* and every support vector known as x_i should be computed (the other parameters α, y, and b are specified from the training phase). Therefore, the hardware/software co-design technique was applied to partition the algorithm efficiently between software and hardware, reaching acceleration of the accumulation of vectors multiplication part via running on hardware.

3.3 The Hardware Implemented Classification System

The structure of the proposed SoC is depicted in Fig. 1, where the HLS IP module is developed first with Vivado HLS tool and then extracted to be implemented as a hardware coprocessor onto the PL part of the Zynq platform. By using the HLS tool with C/C++ language, a top function module was designed as an HLS IP for executing the complicated for loop of the multiplication process. However, the rest of the code was designed as a test bench calling the top function by passing the vectors as formal parameters after reading data of trained SVM model from a file and doing some processing for preparing data.

For optimizing the formal parameters of the top function as being ports of the hardware IP, the AXI4-Stream interface was assigned for the IO interfaces by using the Vivado HLS Directives in order to stream the data of the two vectors for the multiplication into the HLS IP. Also, the AXI-lite bus was assigned for the designed function for controlling the IP core and data flow for the system via communicating with the ARM processor in PS part of the Zynq platform as shown in Fig. 1. The double data type of the vectors was replaced by float to be able to generate an AXI4-Stream interface for streaming data between the ARM CPU in PS and the hardware coprocessor in PL. Also, the source file was changed to a CPP file instead of C as to support HLS stream protocol. As a result, some minor changes in the code were placed using the Vivado HLS tool to satisfy the above reaching high parallelism.

In order to achieve a good level of optimization, some of the different available directives in the HLS tool for optimization were applied for the top module. Pipelining and unrolling the loop for the accumulation of multiplication process were assigned and tested targeting an increase in data throughput rate. Based on synthesis results, **333** clock cycles were required for processing without applying any optimization directives for the loops in the proposed function. However, **169** and **204** clock cycles were required when using unroll and pipeline directives respectively.

Regarding hardware resources utilization estimated from synthesizing, **1514** FF slices, **1794** LUTs and **5** DSP48Es were utilized from using unroll directive, whilst **650** FF slices, **878** LUTs and **5** DSP48Es from using pipeline one. Actually, there is a trade-off between the data throughput and area, as by using the unroll directive more utilization of FF and LUT resources took place than using the pipeline one. However by using the recent Zynq device of large optimized size, the main aim is to increase

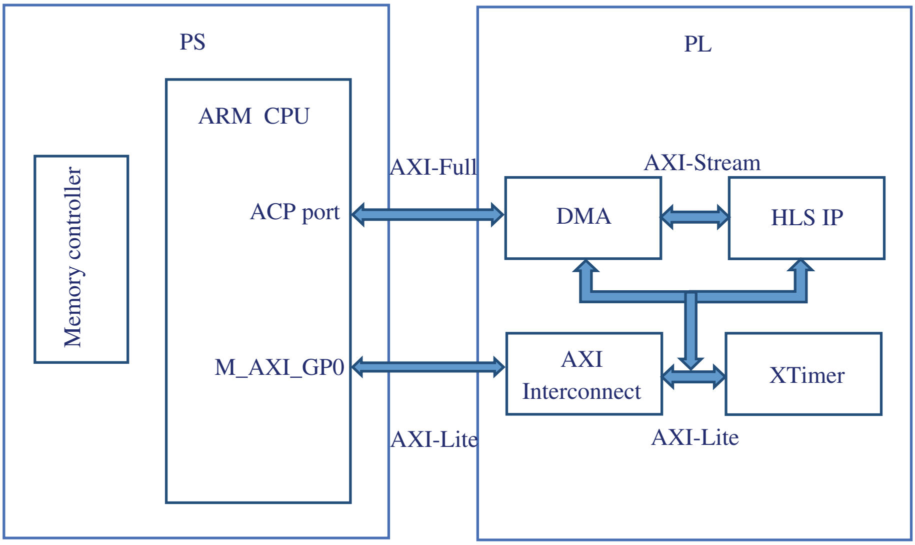

Fig. 1. The block diagram of the proposed system on Zynq

data throughput with reasonable resources utilization. Accordingly, the unrolling technique was exploited for the implementation, reaching throughput improvement.

After successfully synthesizing the code and applying optimization and interfaces directives, the HLS IP core for the classifier was successfully co-simulated and exported as an RTL implementation using the Vivado HLS tool. Then, the exported HLS IP was integrated into a proposed system that was designed by the help of Vivado Design tool as shown in Fig. 2.

The HLS IP core was connected as a Zynq coprocessor to be implemented in the PL with the ARM processor in PS via ACP (Accelerator Coherency Port) using DMA (Direct Memory Access). The DMA controller IP was used to control the data transfer between the ARM processor and the HLS IP through the AXI4-Stream protocol. The ACP is a 64-bit AXI slave interface on the snoop control unit, which provides an asynchronous cache-coherent access point directly from the Zynq PL to the PS with low latency path [35]. Also, an AXI-Timer is instantiated for measuring the number of clock cycles required by the cores for performance comparisons.

Finally after synthesizing, implementing and generating the bitstream of the proposed design using the Vivado tool, it was exported for the Software Development Kit (SDK) to be ready for running the application on Zynq device.

4 Experimental Results

Considering the SVM-light source code as a case study, it was tested first in the Microsoft Visual Studio before using for hardware implementation. The trained model was generated and tested targeting detection of melanoma using a dataset of 356 clinical images (168 melanoma images). The trained data consists of feature extracted

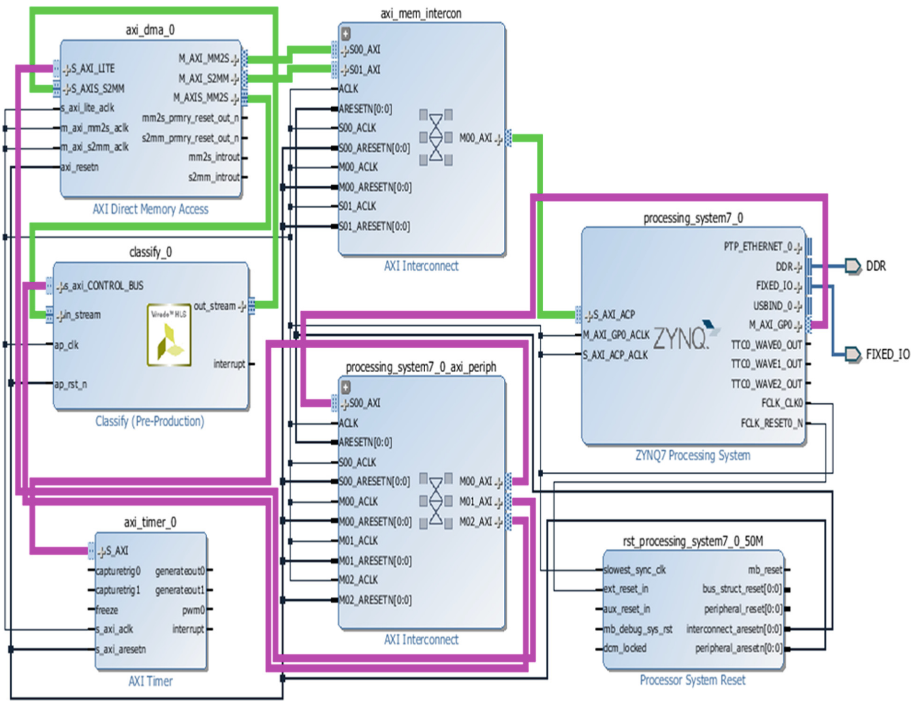

Fig. 2. The implemented system on Zynq

data based on HSV color channels after applying some of the pre-processing and segmentation schemes for the images, with maximum number of features equal to 27 [4]. The default linear kernel function and default parameters for the SVM-light program were used for the training and classification phases. The trained model generated was exploited by the classification phase for the hardware implementation.

The recent Xilinx Vivado 2015.1 Design Suite was used for implementing the proposed system on the Xilinx XC7Z020CLG484-1 target device of the Zynq-7 ZC702 Evaluation Board. An application of classifying a test image was written in C as a test bench to run the exported implementation on the Zynq device by the aid of the Xilinx SDK tool. The experimental results for the hardware accelerator were the same as the software classification results, which showed that the classification accuracy level is preserved without any reduction from the hardware acceleration in contrast to some previous work in literature [13, 17].

In addition, the AXI Timer IP core was used to compare the total computing time between running the designed code for the accumulation of multiplication process on ARM processor in PS and on hardware downloaded in PL part. The ARM processor was running at 666.67 MHz, whilst the computing logic was running at 50 MHz. The total number of clock cycles for the software on ARM processor was **64998**, while for the AXI DMA and hardware accelerator it was **549** cycles. Accordingly, the acceleration factor was greater than **110**, which showed significant acceleration from the implemented hardware accelerator for the proposed SVM classification system.

Table 1 summarizes the hardware resources utilization for the implementation of the proposed system on the Zynq device. It is clear that the percentages of the resources utilization are very low, showing significant improvement in area savings.

Moreover, the power consumption results reported by Vivado tool are detailed in Table 2. The on-chip total power consumption of **1.738** watts is considered to be small reasonable value, meeting embedded systems constraints. The device static power consumes 9 % of the total, whilst the remainder 91 % was dissipated by the dynamic activity mostly from the Zynq PS component compared to other on-chip components.

Interestingly, a hardware/software co-design method of SVM classification of melanoma images is to be considered as an added work in literature that was realized on the recent hybrid Zynq platform using the HLS design methodology. In addition, our proposed implementation achieved lower hardware resources utilization than some of the previous FPGA-based SVM classification implementations of different applications in literature [10, 12, 21, 23, 25, 34, 36]. Concerning power consumption, our implementation demonstrated lower power dissipation than other related work [6, 12, 25]. Therefore, our proposed system could be deployed as a real-time embedded system dedicated for melanoma detection.

Table 1. Device utilization summary.

Resource	Utilization	Available	Utilization %
Slice FF Registers	5584	106400	5.25
Slice LUTs	4373	53200	8.22
Memory LUT	173	17400	0.99
BRAM	3	140	2.14
DSP48	5	220	2.27
BUFG	1	32	3.13

Table 2. On-Chip components power consumption summary.

On-Chip Component	Power (mW)
Clocks	9
Logic	3
Signals	4
BRAM	2
DSPs	<1
PS7	1565
Total Dynamic Power	1584
Device Static Power	154
Total On-Chip Power	1738

5 Conclusion

A hardware/software co-design was realized for the SVM classification of melanoma images on the hybrid Zynq platform. The system, implemented on a recent Zynq device, achieved higher throughput and significant savings in area and power consumption. The implemented SVM classifier was significantly accelerated by hardware, preserving the classification accuracy without any loss. Additionally by using the HLS design methodology, the development time and effort was reduced and embedded system design was simplified.

Hardware implementation results of higher performance, lower resources utilization and less power consumed meet the embedded systems constraints. Consequently, a real time embedded system for efficient melanoma images classification could be realized by the proposed system, achieving high performance and low cost.

For future work, the proposed embedded system of the SVM classifier could be applied for different classification problems. Also, additional work on hardware architectures would be explored targeting an increase in the SVM classification accuracy. Additionally, the proposed hardware/software co-design realized on Zynq with the HLS design method could be adopted by hardware designers for implementing their embedded systems. Moreover, other optimization techniques like DPR technique could be employed for further design optimizations.

Finally, the hardware implementation on FPGA could be extended in the future for other computationally demanding parts in the process, aiming to reach an efficient real-time CAD system for enhancing early detection of melanoma with high performance and low cost.

References

1. Sathiya, S.B., Kumar, S.S., Prabin, A.: A survey on recent computer-aided diagnosis of melanoma. In: International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp. 1387–1392. IEEE Press (2014)
2. Véstias, M.P.: High-performance reconfigurable computing granularity. In: Encyclopedia of Information Science and Technology, pp. 3558–3567 (2015)
3. Saegusa, T., Maruyama, T., Yamaguchi, Y.: How fast is an FPGA in image processing? In: International Conference on Field Programmable Logic and Applications, FPL 2008 pp. 77–82. IEEE Press (2008)
4. Sabouri, P., GholamHosseini, H., Larsson, T., Collins, J.: A cascade classifier for diagnosis of melanoma in clinical images. In: 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 6748–6751. IEEE Press (2014)
5. Nayak, J., Naik, B., Behera, H.: A comprehensive survey on support vector machine in data mining tasks: applications & challenges. Int. J. Database Theory Appl. **8**, 169–186 (2015)
6. Patil, R., Gupta, G., Sahula, V., Mandal, A.: Power aware hardware prototyping of multiclass SVM classifier through reconfiguration. In: 2012 25th International Conference on VLSI Design (VLSID), pp. 62–67. IEEE Press (2012)
7. Hussain, H.M., Benkrid, K., Seker, H.: Reconfiguration-based implementation of SVMclassifier on FPGA for classifying microarray data. In: Proceedings of the Annual InternationalConference of the IEEE Engineering in Medicine and Biology Society (EMBS), pp. 3058–3061 (2013)
8. Hussain, H., Benkrid, K., Seker, H.: Novel dynamic partial reconfiguration implementations of the support vector machine classifier on FPGA. Turkish Journal of Electrical Engineering and Computer Sciences (2014)

9. Mandal, B., Sarma, M.P., Sarma, K.K.: Implementation of systolic array based SVM classifier using multiplierless kernel. In: Proceedings of the 16th International Conference on Automatic Control, Modelling & Simulation (ACMOS 2014), pp. 288–294. WSEAS Press (2014)
10. Kyrkou, C., Theocharides, T.: A parallel hardware architecture for real-time object detection with support vector machines. IEEE Trans. Comput. **61**, 831–842 (2012)
11. Papadonikolakis, M., Bouganis, C.: Novel cascade FPGA accelerator for support vector machines classification. IEEE Trans. Neural Netw. Learn. Syst. **23**, 1040–1052 (2012)
12. Kyrkou, C., Theocharides, T., Bouganis, C.-S.: An embedded hardware-efficient architecture for real-time cascade support vector machine classification. In: 2013 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIII), pp. 129–136. IEEE Press (2013)
13. Kyrkou, C., Bouganis, C.-S., Theocharides, T., Polycarpou, M.M.: Embedded hardware-efficient real-time classification with cascade support vector machines. IEEE Trans. Neural Netw. Learn. Syst. **27**, 99–112 (2015)
14. Shuai, X., Yibin, L., Zhiping, J., Lei, J.: Binarization based implementation for real-time human detection. In: 2013 23rd International Conference on Field Programmable Logic and Applications (FPL), pp. 1–4 (2013)
15. Ruiz-Llata, M., Guarnizo, G., Yébenes-Calvino, M.: FPGA implementation of a support vector machine for classification and regression. In: The 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1–5. IEEE Press (2010)
16. Jallad, A.H.M., Mohammed, L.B.: Hardware Support Vector Machine (SVM) for satellite on-board applications. In: 2014 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), pp. 256–261 (2014)
17. Kyrkou, C., Theocharides, T., Bouganis, C.S.: A hardware-efficient architecture for embedded real-time cascaded support vector machines classification. In: Proceedings of the 23rd ACM international conference on Great lakes symposium on VLSI, pp. 341–342. ACM (2013)
18. Papadonikolakis, M., Bouganis, C.-S.: A novel FPGA-based SVM classifier. In: International Conference on Field-Programmable Technology (FPT), pp. 283–286. IEEE Press (2010)
19. Kim, S., Lee, S., Cho, K.: Design of high-performance unified circuit for linear and non-linear SVM classifications. J. Semicond. Technol. Sci. **12**, 162–167 (2012)
20. Koide, T., Anh-Tuan, H., Okamoto, T., Shigemi, S., Mishima, T., Tamaki, T., Raytchev, B., Kaneda, K., Kominami, Y., Miyaki, R., Matsuo, T., Yoshida, S., Tanaka, S.: FPGA implementation of type identifier for colorectal endoscopie images with NBI magnification. In: 2014 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), pp. 651–654 (2014)
21. Ago, Y., Nakano, K., Ito, Y.: A classification processor for a support vector machine with embedded DSP slices and block RAMs in the FPGA. In: 2013 IEEE 7th International Symposium on Embedded Multicore Socs (MCSoC), pp. 91–96 (2013)
22. Mizuno, K., Terachi, Y., Takagi, K., Izumi, S., Kawaguchi, H., Yoshimoto, M.: Architectural study of HOG feature extraction processor for real-time object detection. In: 2012 IEEE Workshop on Signal Processing Systems (SiPS), pp. 197–202 (2012)
23. Komorkiewicz, M., Kluczewski, M., Gorgon, M.: Floating point hog implementation for real-time multiple object detection. In: 2012 22nd International Conference on Field Programmable Logic and Applications (FPL), pp. 711–714. IEEE Press (2012)

24. Liu, C., Qiao, F., Yang, X., Yang, H.: Hardware acceleration with pipelined adder for support vector machine classifier. In: 2014 Fourth International Conference on Digital Information and Communication Technology and it's Applications (DICTAP), pp. 13–16. IEEE Press (2014)
25. Pietron, M., Wielgosz, M., Zurek, D., Jamro, E., Wiatr, K.: Comparison of GPU and FPGA implementation of SVM algorithm for fast image segmentation. In: Kubátová, H., Hochberger, C., Daněk, M., Sick, B. (eds.) ARCS 2013. LNCS, vol. 7767, pp. 292–302. Springer, Heidelberg (2013)
26. Kryjak, T., Komorkiewicz, M., Gorgon, M.: FPGA implementation of real-time head-shoulder detection using local binary patterns, SVM and foreground object detection. In: 2012 Conference on Design and Architectures for Signal and Image Processing (DASIP), pp. 1–8 (2012)
27. Qasaimeh, M., Sagahyroon, A., Shanableh, T.: FPGA-based parallel hardware architecture for real-time image classification. IEEE Trans. Comput. Imag. **1**, 56–70 (2015)
28. Kelly, C., Siddiqui, F.M., Bardak, B., Woods, R.: Histogram of oriented gradients front end processing: an FPGA based processor approach. In: 2014 IEEE Workshop on Signal Processing Systems (SiPS), pp. 1–6. IEEE Press (2014)
29. Vivado High-Level Synthesis, http://www.xilinx.com/products/design-tools/vivado/integration/esl-design.html
30. Zynq-7000 All Programmable SoC, http://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html
31. Vivado Design Suite, http://www.xilinx.com/products/design-tools/vivado.html
32. Ning, M., Shaojun, W., Yeyong, P., Yu, P.: Implementation of LS-SVM with HLS on Zynq. In: 2014 International Conference on Field-Programmable Technology (FPT), pp. 346–349. IEEE Press (2014)
33. Joachims, T.: Making large-scale SVM learning practical. In: Schölkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods: Support Vector Learning. MIT Press, Cambridge (1999)
34. Mahmoodi, D., Soleimani, A., Khosravi, H., Taghizadeh, M.: FPGA simulation of linear and nonlinear support vector machine. J. Softw. Eng. Appl. **4**, 320–328 (2011)
35. Zynq-7000 All Programmable SoC Acccelerator for Floating-Point Matrix Multiplication using Vivado HLS, http://www.xilinx.com/support/documentation/application_notes/xapp1170-zynq-hls.pdf
36. Qasaimeh, M., Sagahyroon, A., Shanableh, T.: FPGA-based parallel hardware architecture for real-time image classification. IEEE Trans. Comput. Imag. **1**, 56–70 (2015)