MSDF-SVM: Advantage of Most Significant Digit First Arithmetic for SVM Realization

Saeid Gorgin[1], Mohammadreza Najafi[1], Mohammad H. Golamrezaei[1], Jeong-A Lee[1], Miloˇs D. Ercegovac[2]

[1] Department of Computer Engineering, Chosun University, Gwangju, Republic of Korea

[2] Computer Science Department, University of California, Los Angeles

gorgin@chosun.ac.kr, gholamrezaei@chosun.kr, najafi@chosun.ac.kr, jalee@chosun.ac.kr, milos@cs.ucla.edu

***Abstract*— The implementation of Support Vector Machine (SVM) for processing high-dimensional datasets and achieving both accuracy and fast response times poses substantial challenges, especially with the growing size of modern datasets and feature dimensions. To address these challenges, this paper introduces MSDF-SVM, a novel approach that exploits the parallelism and digit-level pipelining capabilities of FPGA devices and leverages online arithmetic. The proposed method takes advantage of an efficient dot-product unit for computing distances to support vectors and then detects their signs. This process is done on incoming data samples in a serial manner, which reduces the area and memory requirements for each classifier instance. This allows for the optimal utilization of parallel instances based on the target FPGA devices. MSDF-SVM dynamically terminates unnecessary computations when detected, resulting in a considerable average reduction of over 65.6% in energy usage. Notably, this work represents the first effective utilization of online arithmetic to design an SVM accelerator. Experimental evaluations on real-world applications, including voice gender detection and emotion classification, demonstrate the remarkable performance improvements and reduced energy consumption achieved by MSDF-SVM compared to previous designs.**

Keywords*— *Machine Learning, Hardware Accelerator, Support Vector Machine, Computer Arithmetic.

I. Introduction

Machine learning (ML) has become integral to various domains, including computer science, finance, agriculture, and medical sciences [1]. Among the supervised learning algorithms, Support Vector Machine (SVM) is widely used for classification and regression tasks and has shown superior accuracy compared to other popular algorithms like Artificial Neural Networks [1]. SVM achieves linear classification of data by selecting the line with the most reliable margin between data points in each class. The optimal line equation is obtained through Quadratic Programming methods used for solving constrained problems [2]. One of the major challenges in this area is the volume of computations required facing today's large-size and high-dimension datasets, as well as the vast amount of data produced daily. To mitigate this severe obstacle, some hardware accelerators have been offered in recent years. Acceleration helps machine learning algorithms run novel applications, carry out time-consuming tasks faster, and improve resource efficiency [3], [4]. In addition, domain-specific hardware acceleration can be beneficial for running machine learning algorithms on battery-powered devices, especially for the SVM, which is broadly employed in wearable healthcare gadgets [5]. Hence, accelerated computing is always needed because of increasing demands for AI-powered applications in modern lifestyles, intelligent applications, and businesses [6].

In this paper, we focus on proposing a hardware accelerator for the SVM machine learning algorithm to accelerate the process of classification using the most digit first (MSDF) arithmetic on FPGA devices. The proposed method utilizes an efficient dot-product unit [7] for computing distances to support vectors. Employing MSDF arithmetic, our proposed design can discard unnecessary computation and exploit the advantage of massive parallelism in FPGAs to speed up the time-consuming classification process in the SVM. Considering the large-size and high-dimensional today's dataset, the importance of our proposed method becomes more significant and evident. Our contribution and novelties can be summarized as (a) designing an FPGA-based hardware accelerator for SVM using MSDF arithmetic, (b) implementing an early termination mechanism to discard unnecessary computations once they are detected, (c) improving the performance of SVM classification.

The rest of this paper is as follows. Section II describes the background and literature review of this field. Our proposed design and its details are elaborated in Section III, while Section IV provides two real-world applications as a case study. In Section V, we provide a deep evaluation of our proposed design and the best previous one, and finally, the paper is concluded by Section VI.

II. Background and Literature Review

SVM is one of the well-established algorithms in machine learning in various domains, such as face detection, text, and hypertext categorization to classify documents into different groups [8], and image classification [9]. SVM provides better search accuracy for image classification compared to traditional query-based searching techniques and even artificial neural networks in some cases. SVM is also employed in bioinformatics for protein classification, cancer classification, as well as identifying the classes of genes [10]. Besides, SVM has widely been used for recognizing handwritten characters and vehicle classification [11]. SVM uses a technique called kernel trick to convert data; then, based on this conversion, SVM finds the optimal boundary between possible outputs. One of the main pros of the support vector machine method, which makes it very popular, is the ability to solve various problems without losing accuracy by enhancing the size and dimension of the problem.

Whereas the SVM has offered many advantages, including being effective in high dimensional spaces and cases where the number of dimensions is higher than samples, it severely suffers from a computationally intensive nature with time complexity of $O(n^3)$. In recent years, some custom kernel functions have been proposed to mitigate this challenge by leveraging the advantage of the versatile capability of SVM. This capability

makes it possible to specify custom kernels, such as Sigmoid, Gaussian, Hyperbolic Tangent, and Radial Basis Function (RBF) [12]. The linear kernel is the most frequent type of SVM, which is typically used to deal with linearly separable problems, such as facial expression classification [13] and colorectal cancer detection [14]. The majority of customized kernels, such as sigmoid, RBF, and polynomial, can be implemented using linear equations. These types of SVM have widely been employed in addressing real-world problems, e.g., linear Sigmoid for language recognition [15], linear RBF polynomial for pedestrian detection [16]. This is the underlying reason that we have focused on mitigating the challenge of the computationally-intensive nature of linear SVM.

A. *Support Vector Machine Internals*

The SVM is a supervised classifier that finds hyperplanes in n-dimensions space to perform non-overlapping segmentation or regression utilizing all features. It allows a global classification model based on linear discrimination with maximum margin, not considering dependencies between attributes. Intuitively, the hyperplane that best separates data points is the case where the distance to the nearest training data is the largest, as indicated in Fig. 1.

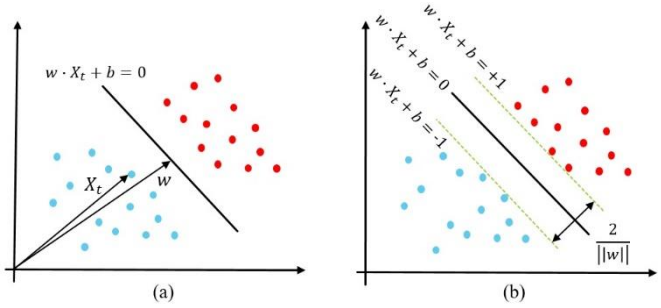

Fig. 1 The procedure of SVM (a) computing distances and (b) determining hyperplanes.

To determine the label (i.e., Y_{X_t}) of test data, Equation 1, or its expanded form in Equation 2, is used. In this equation, if the value of $f(X_t)$ is negative, the Sign function returns the value of -1 as the label; otherwise, if the value of $f(X_t)$ is positive, the label would be $+1$.

$$Y_{X_t} = \text{Sign}\big(f(X_t)\big) \quad (1)$$

$$f(X_t) = \left(\sum_{i=1}^{n}\big(y_i \alpha_i K(X_t, x_i)\big) + b\right) \quad (2)$$

In Equation 2, x_i is the train data, X_t is the test data, and y_i indicates the label of each train data point. The values of α_i and b are calculated from the training phase and $K(X_t, x_i)$ is the kernel function, which for linear one is shown by Equation 3.

$$K(X_t, x_i) = X_t \cdot x_i \quad (3)$$

B. *Most Significant Digit First Arithmetic*

Online arithmetic, as a type of Most Significant Digit First (MSDF) arithmetic, is a form of serial computation that prioritizes processing the most significant digits [17]. Unlike conventional addition and multiplication arithmetic operations, where the result is generated from right to left (least significant positions to most significant positions), Online arithmetic provides the result of the most significant position first. This approach allows for the production of result digits after receiving a limited number of digits, represented as the online delay (δ), while the remaining input components continue to arrive [18]. Consequently, even dependent operations can be executed in an overlapping manner, enabling digit-level massive parallelism.

The aforementioned facility is provided via Online arithmetic because of utilizing the power of redundant number representation, which enables it to handle various representations of a given value. Operating in a serial manner, this approach effectively reduces the area of the arithmetic unit, decreases memory usage, and keeps interconnects to a minimum. Moreover, Online arithmetic is particularly well-suited for computations that require variable precision. Once the desired precision is reached or a specific condition is fulfilled, the current operation can be stopped, allowing the next operation to begin. This feature has a substantial impact on both power consumption and performance [19]

III. Proposed Design

This section describes our proposed approach and its implementation in detail. The main goal pursued in our design is to improve the performance, as well as reduce energy consumption and resource utilization compared to state-of-the-art methods.

In Equation 2, y_i and x_i are training samples, and the value of α_i and b are determined in the training phase. Considering Equation 3, it can be re-written in the form of Equation 4.

$$Y_{X_t} = \text{Sign}(wX_t + b) \quad (4)$$

where w indicates the weight and has a fixed value during the test phase for all data points so that it can be computed during the pre-processing phase.

For hardware acceleration of the inference phase, we have used a hardware-software co-design approach that takes advantage of specific custom hardware based on MSDF dot-product unite [7] beside the embedded processor of the FPGA fabric. The abstract view of the proposed architecture is indicated in Fig. 2. All the interfacing of the components and the read/write operations from/to the memory are handled by the AXI interconnect. Due to independence between required operations for each sample data in the SVM algorithm to classify a large number of sample data, we can exploit massive parallelism over all the samples where an instance of the classifier is replicated several times. The amount of parallelism is defined by the available resources of the target FPGA and also application constraints.

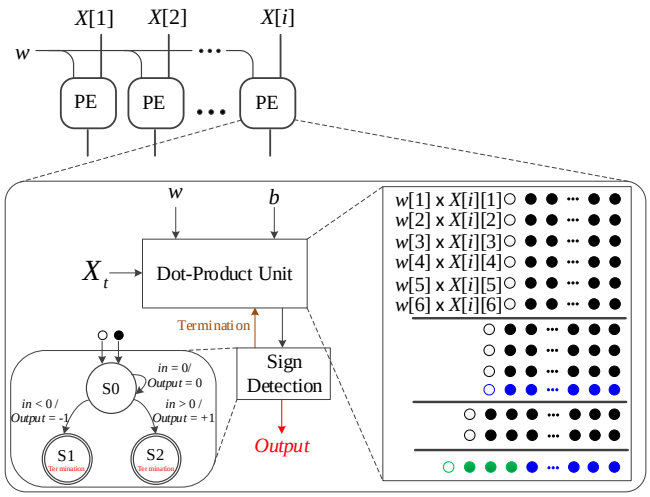

Fig. 2 The abstract view of the MSDF-SVM architecture beside the MSDF dot-product unit architecture with dot-notation clarification ($dim = 6$).

In this architecture, the dot-product of precomputed value (w) and each sample are calculated, and the classification is done based on the result's sign. The value of all dimensions of w are available in parallel, and the value of sample data comes serially from the most significant digit. The results of the bit-serial multiplications are added via an adder tree. Due to the fast carry chain hardware inside the FPGA fabric, in this design, we use carry-propagate adders at each level of the reduction tree. While in ASIC design, carry-save addition provides the best result. The dot-product output (i.e. represented in Binary Signed-Digit) is connected to the Sign Detection unit. In this unit, the leading zero values are ignored, and the classification result is determined by receiving the first non-zero value. Upon receiving the first non-zero value, the label of the classification result (i.e., $+1$ or -1) is sent as the output of the classifier instance since just the sign of dot-product output is enough for classification based on the SVM algorithm. Besides the output, a termination signal is also generated. After raising the termination signal, the classifier instance stops the current computation and starts processing a new data sample. This early termination provides significant performance improvement and meaningfully reduces power and energy consumption. These operations are applied in a pipelined digit serial manner and sweep all the test data points. Due to the independence between required operations for each sample data in the SVM algorithm to classify a large number of sample data, we can exploit massive parallelism over all the samples where an instance of the classifier is replicated several times.

To convey a comprehensive understanding of the internal process of MSDF-SVM, we have provided two numerical examples, i.e. Example 1 and Example 2. The former is classified with negative labels and the latter is used to classify positive labels. The value of w, bias, and test data points for the above-mentioned examples are shown in Table I. The results of all intermediate signals during running the MSDF-SVM are shown in Table II. For ease of understanding, the same color as Fig. 2 was used for representing signals. Moreover, a few points should be considered: (1) all Negabits are represented in IEN, (2) All the computations are carried out based on MSDF arithmetic (e.g., in cycle #1, Example 1, the most significant position of test data points are "0", so the value of Serial Input is represented as "000000"), (3) The value of Shifted Sum in cycle #1 is zero, however, instead of zero we use the sign-extended value of bias, and finally (4) There is a possibility for early termination of the process, as shown by these two examples, after three cycles classification is terminated instead of processing all bits of test data points.

IV. MSDF-SVM in Real-World Applications

We have implemented our designs on two real-world applications of machine learning, i.e., gender detection and emotion classification, as case studies to indicate the effectiveness and performance of our proposed design.

TABLE I. The Value Of w, $bias$, and Test Data Points.

	$w[1]$	$w[2]$	$w[3]$	$w[4]$	$w[5]$	$w[6]$	***bias***	$X[i][1]$	$X[i][2]$	$X[i][3]$	$X[i][4]$	$X[i][5]$	$X[i][6]$
Example 1	0^-0010	1^-0100	0^-0110	1^-0001	0^-1000	0^-0001	$\mathbf{0^-0101}$	01011	01000	01100	00101	00111	01101
Example 2	0^-0010	0^-0100	0^-0110	0^-1001	0^-1000	1^-1111	$\mathbf{1^-1101}$	10000	11111	10110	11101	10000	00001

TABLE II. The Results Of All Intermediate Signals During Running The MSDF-SVM.

# Cycle	Serial Input	$w[1] \times X[i][1]$	$w[2] \times X[i][2]$	$w[3] \times X[i][3]$	$w[4] \times X[i][4]$	$w[5] \times X[i][5]$	$w[6] \times X[i][6]$	Shifted Sum	Partial Sum	S	z (Output Posibit)	z^- (Output Negabit)	Early Termination
Example 1 (classified as negative label)													
1	000000	1^-0000	1^-0000	1^-0000	1^-0000	1^-0000	1^-0000	$\mathbf{0^-10101}$	$0^-1110101$	0^-111	0	1^-	0
2	111001	0^-0010	1^-0100	0^-0110	1^-0000	1^-0000	0^-0001	1^-01010	$0^-1100111$	0^-110	1	0^-	0
3	001111	1^-0000	1^-0000	0^-0110	1^-0001	0^-1000	0^-0001	1^-01110	$0^-1101110$	0^-110	0	0^-	1
Example 2 (classified as positive label)													
1	111110	1^-0010	1^-0100	1^-0110	1^-1001	1^-1000	1^-0000	$\mathbf{1^-01101}$	$1^-0101010$	1^-010	0	1^-	0
2	010100	1^-0000	1^-0100	1^-0000	1^-1001	1^-0000	1^-0000	1^-10100	$1^-0100001$	1^-010	0	1^-	0
3	011100	1^-0000	1^-0100	1^-0110	1^-1001	1^-0000	1^-0000	1^-00010	$1^-0010101$	1^-001	1	1^-	1

A. *Voice Gender Detection*

Gender detection is vital in real-time applications, but accurately distinguishing genders from images and voices remains challenging. Human voices carry key communication features, while variations in accents add complexity for machines. To address this, we utilize MSDF-SVM, incorporating early termination for faster execution and reduced resource consumption. Experiments with [20] audio data validate the effectiveness of our approach in enhancing gender detection.

B. *Emotion Classification*

We applied our proposed design to emotion classification, an essential task with significant impact in various domains. Emotion recognition poses challenges due to its abstract nature compared to visual semantics [21]. It is computationally intensive, particularly for resource-limited embedded systems and edge devices. To address this, we utilized our SVM accelerator with EEG data [22], featuring around two thousand features. Our real-world evaluation demonstrated improved performance and fair comparisons against existing methods.

V. Evaluation and Comparison

This section evaluates and compares our proposed design with state-of-the-art methods. We conducted experiments on Zynq UltraScale+ XCZU7EV MPSoC Xilinx device (ZCU104 Board) and included previous designs for a valid and fair comparison.

A. *Experimental Setup*

We selected the best state-of-the-art implementation of linear SVM on FPGA [23] (i.e., SVM-Parallel). This implementation encounters a challenge in processing high-dimensional datasets, where the required hardware exceeds available FPGA resources. For instance, in the Emotions dataset [22], which has more than two thousand features, the required hardware resources for SVM-Parallel implementation pass the available budget. This is the main reason that the figures for *Speedup* and *Energy Consumption* for this design are not reported in some parts of the evaluation. To tackle this obstacle, we took advantage of time multiplexing and hardware re-using techniques in SVM-MAC, where the required hardware resources are sufficient. After implementing our proposed and SVM-Parallel using VHDL, we selected SVM-MAC as the baseline model and two different datasets, including Voices [20] and Emotions [22], for conducting a fair and valid comparison. The properties and characters of these datasets are reported in Table III.

TABLE III. The Properties of Datasets Used for The Evaluation.

Dataset	# Features	#Test Samples	#Training
Voice	20	634	2536
Emotions	2509	284	1136

B. *Comparison of Resource Utilization*

Resource efficiency is essential aspect of our design. A comparison between our proposed design and other related works regarding resource usage has been reported in Table IV to indicate the improvement in our design. As indicated in Table IV, our proposed design consumed the lowest resources in terms of the number of LUT, and the number of Flip-Flops (FF) compared to the other designs running both datasets. This improvement would be more critical when the SVM classifier needs to be run on edge devices and portable devices with limited resources and areas.

TABLE IV. Comparison Between Our Proposed Methods and Previous Related Works

Design	Dataset	#LUT	#FF	#PE
SVM-MSDF	Voice	132172	7640	191
	Emotions	132520	47	1
SVM-MAC	Voice	132264	2816	88
	Emotions	132264	2816	88
SVM-Parallel Tree	Voice	150015	160	5
	Emotions	-	-	-

C. *Comparison of Speedup*

Speedup is the most important factor in our evaluation since the main aim of our proposed design was to improve the speed of SVM classification, especially when it faces novel large-size datasets. We have measured the amount of *Speedup* for our design by running two Voice and Emotions datasets. The results of this comparison are reported in Fig. 3. Our proposed design was placed at the first rank compared to the state-of-the-art related design on both Voice and Emotions datasets. In the gender detection problem, our design indicated 5X improvements in *Speedup* against the SVM-MAC and SVM-Parallel Tree. This improvement was 2X in solving the problem of emotions classification. Also, as mentioned before, implementing SVM-Parallel was not possible for this experiment on the Emotions dataset.

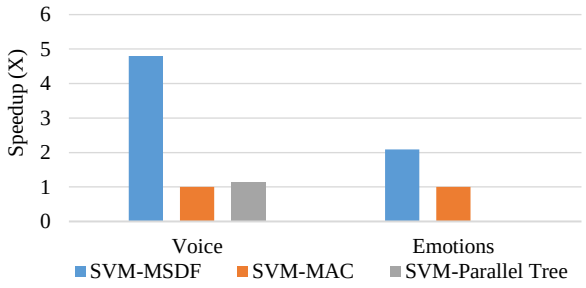

Fig. 3 Comparison of *Speedup* between the proposed design and previous related works.

D. *Comparison of Energy Consumption*

Another aspect of comparison in our paper is *Energy Consumption*, which is highly important for modern smart and portable devices. We have measured the improvements attained for each design against the base implementation. Fig. 4 illustrates the results of this evaluation. As depicted in Fig. 4, our proposed design exhibits superior performance in terms of Energy Consumption when compared to the state-of-the-art approach. Notably, the static power consumption remains consistent with a value of 0.594 across all designs during the experiments. It can be inferred from the results that our design achieves better Speedup, Resource Utilization, and Energy Consumption than other relevant designs in both voice gender detection and EEG emotion classification experiments.

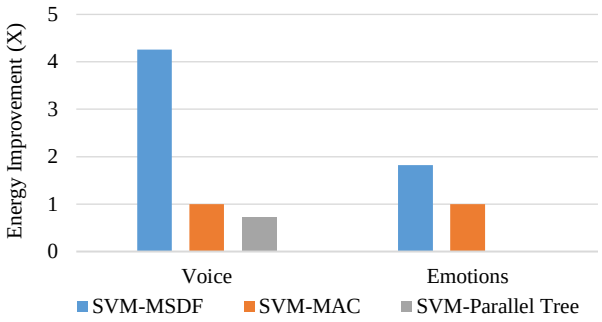

Fig. 4 Comparison of improvement in terms of *Energy Consumption* for each design.

VI. Conclusion

Support vector machine is one of the most well-established machine learning algorithms, which has widely been employed to address classification and regression problems in a wide range of applications. In this paper, we proposed a novel hardware accelerator for the SVM to mitigate its computationally-intensive nature and improve its performance and efficiency in facing huge datasets. Our proposed approach leveraged Online arithmetic and massive parallelism via digit-level pipelining to allow an early termination mechanism, which can early detect and neglect unnecessary computations. Our design was implemented on an FPGA using VHDL, where two real-world applications, i.e., voice gender detection and emotion classification, were employed for evaluating the proposed design. Our proposed design significantly improved the performance of the SVM in terms of *Speedup*, *Resource Utilization*, and *Energy Consumption*. The results of the comparison also proved that our proposed approach could surpass the related works and the state-of-the-art designs. Considering the sheer amount of data produced daily, the importance of our proposed design becomes more significant. Also, our proposed accelerator can be effectively deployed in various scenarios where the SVM faces large-size datasets or has to be run on resource-limited devices, like wearable equipment and edge devices.

Acknowledgment

This research was supported by Brain Pool program funded by the Ministry of Science and ICT through the National Research Foundation of Korea (NRF-2021H1D3A2A02040040) and in part supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2020R1I1A3063857).

References

[1] S. Chidambaram and K. G. Srinivasagan, "Performance evaluation of support vector machine classification approaches in data mining," Cluster Comput., vol. 22, no. 1, pp. 189–196, 2019, doi: 10.1007/s10586-018-2036-z.

[2] C. A. Floudas and V. Visweswaran, "Quadratic Optimization BT - Handbook of Global Optimization," R. Horst and P. M. Pardalos, Eds. Boston, MA: Springer US, 1995, pp. 217–269.

[3] S. Gorgin, M. Gholamrezaei, D. Javaheri, and J.-A. Lee, "kNN-MSDF: A Hardware Accelerator for k-Nearest Neighbors Using Most Significant Digit First Computation," *in 2022 IEEE 35th International System-on-Chip Conference (SOCC)*, 2022, pp. 1–6, doi: 10.1109/SOCC56010.2022.9908102.

[4] S. Gorgin, M. Gholamrezaei, D. Javaheri and J. -A. Lee, "An Energy-Efficient K-means Clustering FPGA Accelerator via Most-Significant Digit First Arithmetic," *in 2022 International Conference on Field-Programmable Technology (ICFPT)*, Hong Kong, 2022, pp. 1-4.

[5] M. Kee and G.-H. Park, "A Low-Power Programmable Machine Learning Hardware Accelerator Design for Intelligent Edge Devices," *ACM Trans. Des. Autom. Electron. Syst.*, Apr. 2022, doi: 10.1145/3531479.

[6] H. Park and S. Kim, "Chapter Three - Hardware accelerator systems for artificial intelligence and machine learning," *in Hardware Accelerator Systems for Artificial Intelligence and Machine Learning*, vol. 122, S. Kim and G. C. B. T.-A. in C. Deka, Eds. Elsevier, 2021, pp. 51–95.

[7] Saeid Gorgin, Mohammad H. Golamrezaei, Jeong-A Lee, and Miloˇs D. Ercegovac, "An Efficient Dot-Product Unit Based on Online Arithmetic for Variable Precision Applications," *will be appear in 2023 57th Asilomar Conference on Signals, Systems, and Computers*.

[8] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," Neurocomputing, vol. 408, pp. 189–215, 2020, doi: https://doi.org/10.1016/j.neucom.2019.10.118.

[9] Najafi Mohammadreza, Saeid Gorgin, Danial Javaheri, and Jeong-A Lee, "Ice Detection on Edge Device Based on Most Significant Digit First SVM," *in 2022 6th International Conference on Video and Image Processing (ICVIP '22)*. pp. 61–66.

[10] S. Huang, N. Cai, P. P. Pacheco, S. Narrandes, Y. Wang, and W. Xu, "Applications of Support Vector Machine (SVM) Learning in Cancer Genomics," Cancer Genomics Proteomics, vol. 15, no. 1, pp. 41–51, 2018, doi: 10.21873/cgp.20063.

[11] H. Fu, H. Ma, Y. Liu, and D. Lu, "A vehicle classification system based on hierarchical multi-SVMs in crowded traffic scenes," Neurocomputing, vol. 211, pp. 182–190, 2016, doi: https://doi.org/10.1016/j.neucom.2015.12.134.

[12] S. Afifi, H. GholamHosseini, and R. Sinha, "FPGA Implementations of SVM Classifiers: A Review," SN Comput. Sci., vol. 1, no. 3, p. 133, 2020, doi: 10.1007/s42979-020-00128-9.

[13] S. Saurav, S. Singh, R. Saini, and A. K. Saini, "Hardware Accelerator for Facial Expression Classification Using Linear SVM BT - Advances in Signal Processing and Intelligent Recognition Systems," 2016, pp. 39–50.

[14] T. Koide et al., "FPGA implementation of type identifier for colorectal endoscopie images with NBI magnification," in 2014 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 2014, pp. 651–654, doi: 10.1109/APCCAS.2014.7032865.

[15] Z. Nie, X. Zhang, and Z. Yang, "An FPGA Implementation of Multi-Class Support Vector Machine Classifier Based on Posterior Probability," 2012.

[16] M. Berberich and K. Doll, "Highly Flexible FPGA-Architecture of a Support Vector Machine," in 45. MPC-Workshop, Albstadt-Sigmaringen, 2014, no. 45, pp. 25–32, [Online]. Available: https://www.mpc-gruppe.de/workshopbaende.

[17] M. D. Ercegovac, "On left-to-right arithmetic," in 2017 51st Asilomar Conference on Signals, Systems, and Computers, 2017, pp. 750–754, doi: 10.1109/ACSSC.2017.8335445.

[18] M. D. Ercegovac and T. Lang, Digital Arithmetic. USA: Elsevier Science Inc., 2004.

[19] M. D. Ercegovac, "On Reducing Module Activities in Online Arithmetic Operations," in 2020 54th Asilomar Conference on Signals, Systems, and Computers, 2020, pp. 524–528, doi: 10.1109/IEEECONF51394.2020.9443576.

[20] I. E. Livieris, E. Pintelas, and P. Pintelas, "Gender Recognition by Voice Using an Improved Self-Labeled Algorithm," Mach. Learn. Knowl. Extr., vol. 1, no. 1, pp. 492–503, 2019, doi: 10.3390/make1010030.

[21] H. Zhang and M. Xu, "Weakly Supervised Emotion Intensity Prediction for Recognition of Emotions in Images," IEEE Trans. Multimed., vol. 23, pp. 2033–2044, 2021, doi: 10.1109/TMM.2020.3007352.

[22] J. J. Bird, D. R. Faria, L. J. Manso, A. Ekárt, and C. D. Buckingham, "A Deep Evolutionary Approach to Bioinspired Classifier Optimisation for Brain-Machine Interaction," Complexity, vol. 2019, p. 4316548, 2019, doi: 10.1155/2019/4316548.

[23] X. Song, H. Wang, and L. Wang, "FPGA Implementation of a Support Vector Machine Based Classification System and Its Potential Application in Smart Grid," in 2014 11th International Conference on Information Technology: New Generations, 2014, pp. 397–402, doi: 10.1109/ITNG.2014.