Extended Abstracts of the 2013 International Conference on Solid State Devices and Materials, Fukuoka, 2013, pp140-141

An Analog VLSI Implementation of One-Class Support Vector Machine

Renyuan Zhang[1], Mineo Kaneko[1]and Tadashi Shibata[2]

[1] School of Information Science, Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
Phone: +81-761-51-1274 E-mail: rzhang@jaist.ac.jp
[2] Center for Innovative Integrated Electronics Systems, Tohoku University

Abstract

The one-class support vector machine classifier is implemented by a fully parallel analog VLSI processor. A proof-of-concept chip is built for the pattern recognition problems of highly dimensional samples such as image feature vectors.

1. Introduction

The support vector machine (SVM) employing Gaussian kernel functions has shown the remarkable performance in the pattern recognition problem of highly dimensional sample vectors [1]. In recent years, many VLSI implementations of SVM have even been reported with some benefits over the processing speed and power consumption [2]. Since the Gaussian function for highly dimensional vectors is very expensive to implement in silicon, the analog VLSI implementations were helpful for the error-tolerant problems such as image processing [3].

In some real-world tasks, these conventional SVM implementations as typical two-class classifiers are difficult to solve the multi-class classification problems. Therefore, the one-class support vector machine (OC-SVM) algorithm or called support vector domain description (SVDD) was developed [4]. By using OC-SVM technology, the data domain for one single class of samples can be efficiently described without any contribution of the "opposite class", which is convenient to expend the number of classes. Another obvious benefit of OC-SVM algorithm is that the test patterns might be reasonably rejected by all the sample classes, which possibly occurs in the real-world applications. So far, this algorithm has been applied in the recognition problems of highly dimensional sample vectors such as images [5]. On the other hand, OC-SVM algorithm is more complex and time-consuming than the conventional SVM mathematically since the directly solving process of OC-SVM requires many computations among matrices [6]. Some efforts for enhancing the processing speed were made to implement OC-SVM for two-dimensional sample vectors by software programs [6, 7]. However, further efforts are needed for solving those problems of highly dimensional vectors (for instance, 64-D feature vectors of images in this work). Some hardware implementations of OC-SVM also suffer from the complexity problem [8].

In this work, an analog VLSI processor implementing OC-SVM algorithm is built for highly dimensional sample vectors. By using the analog fully parallel architecture, the computations among matrices are carried out in real-time. In this manner, the learning operation can be accomplished without any clock-based iteration. The real images, which

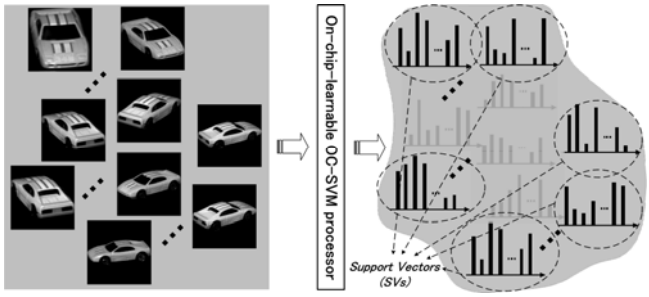

Fig. 1 One-class classification problem for highly dimensional sample vectors (images) is solved by OC-SVM processor.

are pre-transferred into 64-D feature vectors, are employed as learning samples and test patterns. Through the on-chip learning session, the data domain is described within 0.2 micro-seconds. From the measurement results, all the test patterns are correctly recognized by the built chip.

2. Applications and Algorithm in This Work

The image recognition problem is taken as the example shown in Fig.1. Some pre-processed 64-D feature vectors of images are input to the OC-SVM processor. During the on-chip learning session, several support vectors (SVs) are autonomously selected and the data domain is described by them. After learning, the test patterns can be recognized or rejected by this domain description.

Considering the sample vectors $\mathbf{X}_i s$, the task of domain description is to obtain sufficiently compact sphere of the samples' domain as: $(\mathbf{X}_i - \mathbf{a})^{\mathrm{T}}(\mathbf{X}_i - \mathbf{a}) \le R^2 + \xi_i$, where *a* and *R* are the center and volume of this sphere, respectively; and ξ is a slack variable. Employ the Gaussian kernel function: $\mathrm{K}(\mathbf{X}_i, \mathbf{X}_j) = \exp(-\| \mathbf{X}_i, \mathbf{X}_j \|^2 / \sigma)$, the task becomes to pursue α_i minimizing [4]:

$$L = 1 - \sum \alpha_i^2 - \sum_{i \ne j} \alpha_i \alpha_j K(\mathbf{X}_i, \mathbf{X}_j), \qquad (1)$$

when $\sum \alpha_i^2 = 1$. By introducing a transient parameter λ, we solved this quadratic programming problem based on the following learning rule:

$$\begin{cases} \alpha_i \leftarrow \frac{1}{2}(\lambda - \sum_{i \ne j} \alpha_j K(\mathbf{X}_i, \mathbf{X}_j)) \\ \lambda \leftarrow \frac{1}{N}(1 + \sum_i \sum_j \alpha_j K(\mathbf{X}_i, \mathbf{X}_j)) \end{cases}. \qquad (2)$$

This learning operation is automatically carried out on-chip by the OC-SVM processor. Receiving a test pattern vector $\mathbf{Z}$, the decision function is checked to accept or reject $\mathbf{Z}$ as: $1 - 2\sum \alpha_i K(\mathbf{Z}, \mathbf{X}_i) + \sum_{i,j} \alpha_i K(\mathbf{X}_i, \mathbf{X}_j) \le R^2$. Since the constant items can be observed by any one of support vectors $\mathbf{X}_s$, the accept condition becomes:

$$\sum \alpha_i K(\mathbf{Z}, \mathbf{X}_i) \ge \sum \alpha_i K(\mathbf{X}_s, \mathbf{X}_j). \qquad (3)$$

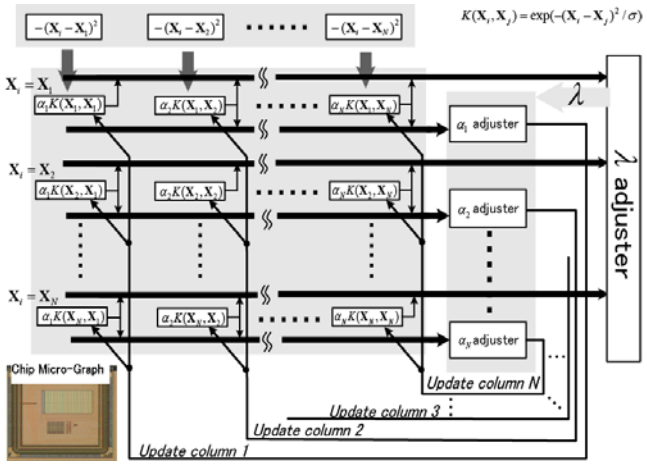

Fig. 2 Architecture of built processor for implementing OC-SVM along with the micro-graph of built chip.

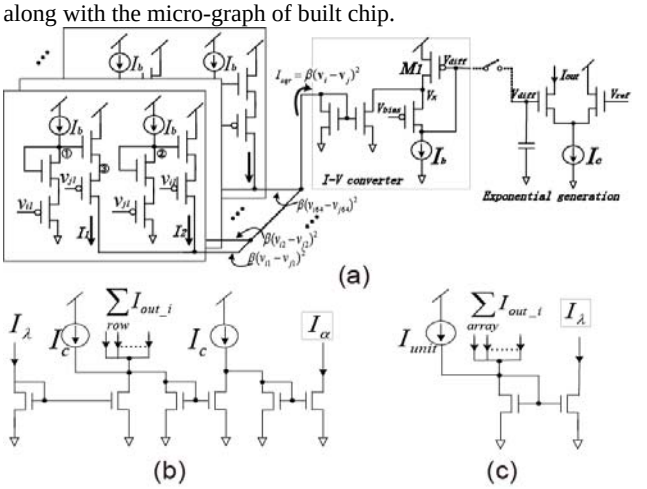

Fig. 3 (a) 64-D Gaussian-generation circuitry [3]; (b) Alpha updating circuit and (c) Lambda updating circuit.

3. Hardware Implementation and Experiments

The architecture of proposed on-chip learnable OC-SVM processor is shown in Fig.2. Both learning and classifying operations are automatically processed in fully parallel. The analog circuitry as shown in Fig.3 (a) is used to compute 64-D Gaussian kernel function [3]. The Alpha and Lambda updating circuits are illustrated in Fig.3 (a) and (b), respectively. The complex computation in Eq.2, which is avoided by software programs [6], can be conveniently carried out by this parallel architecture in real-time.

As a toy-example, the data domain of sixteen samples with two dimensions can be described as it is illustrated in Fig.4. It should be noticed that the feature of domain description is flexible by tuning the parameters of OC-SVM processor easily. A proof-of-concept VLSI chip was fabricated for the recognition problem of highly dimensional vectors. Sixteen image feature vectors with 64 dimensions (from COIL-20 database) are employed as learning samples. As it is shown in Fig.5, the learning operation for domain description is accomplished within 0.2 micro-seconds. From the chip measurement results, all the test patterns are correctly recognized or rejected by the built OC-SVM processor. The arbitrary combination of multiple domain descriptions realizes the multi-class classification without any additional mechanism.

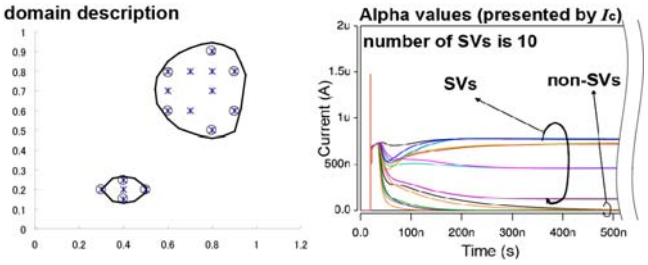

Fig. 4 Domain description for 16 samples with two dimensions.

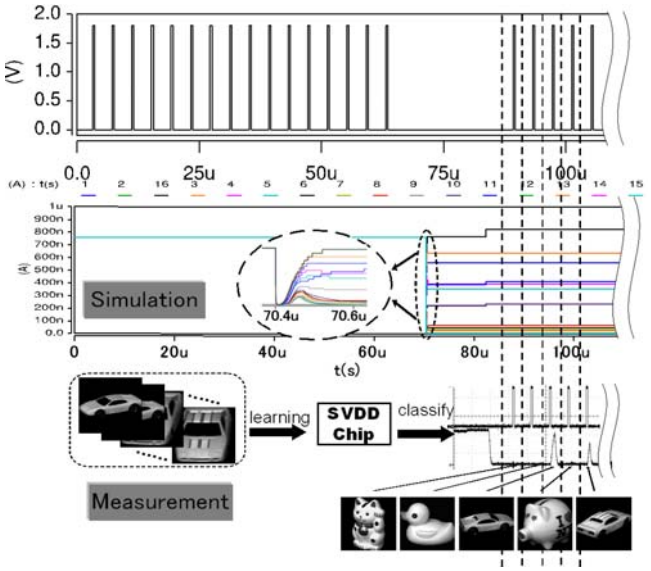

Fig. 5 Pattern recognition of image feature vectors (64-D).

4. Conclusions

The one-class support vector machine (OC-SVM) algorithm was implemented by an on-chip-learnable analog VLSI processor for the recognition problem of highly dimensional sample vectors. Mathematically, the OC-SVM problem was solved in fully parallel, which was friendly carried out by an array of analog circuitries. Through the on-chip learning operation, the data domain for one single class of sample vectors can be described by several support vectors within 0.2 micro-seconds. From the chip measurement results, all the test patterns representing real images were correctly recognized or rejected by the built processor.

References

[1] V. Vapnik, *The Nature of Statistical Learning Theory* (1995).

[2] S. Chakrabartty, and G. Cauwenberghs, IEEE J. Solid-State Circuits **42** (2007) 1169.

[3] R. Zhang and T. Shibata, Jpn. J. Appl. Phys.**51**(2012) 04DE10.

[4] D. M. J. Tax, and R. P. W. Duin, Pattern Recognition Letters **20** (1999) 1191.

[5] M. Katsurai *et al*, *IEEE International Symposium on Communications and Information Technologies* (2010) 991.

[6] H. Wang *et al.*, *Proc. IEEE International Conference on Natural Computation* (2010) 824.

[7] S. Khazai *et al.*, IEEE Geoscience and Remote Sensing Letters **9** (2012) 594.

[8] R. A. Patil *et al.*, *IEEE International Conference on VLSI Design* (2012) 62.