

High Efficiency On-Board Hyperspectral Image Classification with Zynq SoC

Ning Ma^a, Shaojun Wang, Syed Mohsin Ali, Xiuhai Cui and Yu Peng

Department of Automatic Test and Control, Harbin Institute of Technology, Harbin, China

Abstract. Because of the downlink bandwidth bottleneck and power limitation on satellite, the demands for low power cost high performance on-board payload data processing which can reduce the volume of communication data are growing as well. This paper proposes a high efficiency architecture for on-board hyperspectral image classification in a Zynq Soc to achieve real-time performance. The Hamming-distance based Support vector machine (SVM) is adopted to get a high accuracy and low energy consumption for multi-class classification. The sequential control and the computing data path are realized in ARM processor and Programmable logic respectively. By the pipelined computing data path, a satisfying speedup is reached and thus lowers the energy consumption. The experiments on real hyperspectral image datasets demonstrate that our architecture can achieve 97.8% overall accuracy, 2.5~330x speed up and 11~835x energy saving compared with different state-of-art embedded platforms. For the AVIRIS spectrometer in real NASA application, it can realize real-time image classification.

1 Introduction

The spatial, spectral and temporal resolutions of hyperspectral image (HSI) in remote sensing keep increasing in recent years. However, the limited downlink bandwidth between satellites and ground station cannot catch up with the drastically increased data volume from high resolution spectroscopy instruments [1]. One approach to resolve this downlink bottleneck is to process the image data onboard and in real-time to decrease the data size to be transferred by downlink [2]. This requires the onboard processing system not only to achieve high performance to process the online continuous image data, but also to meet the harsh limitation of satellite, such as power consumption, size and weight.

In recent years, significant efforts have been made to cope with above challenges. In [3], Graphics Processing Units (GPUs) was used to process hyperspectral images (HSI) and to gain a superb performance while it is difficult to employ it for onboard satellite applications because of its high power consumption. Due to the superiority of FPGA devices for onboard application, Bernabe et al. [4] developed an automatic target generation process system for HSI using FPGAs.

Classification is one of main tasks for remote sensing image processing. Support vector machine (SVM) algorithm which can perform equal or better accuracy than other classifiers is widely used for hyperspectral classification [5]. It has a number of advantages over its counterparts, such as asymmetry of computation in training and classification, generalization capacity and

good performance on small training data set [6]. HSI classification is multi classification problem, and different strategies to build a multi-classifier by SVM binary classifier has been developed in [7]. Hamming-distance SVM is the most potential choice by overcoming deficiencies of other methods [8].

A lot of researchers have already worked on SVM implementation utilizing different hardware. A. H. M. Jallad et al. [9] proposed a binary SVM classifier on SRAM based FPGA which was designed to identify the Cloud and Non-Cloud pixels. However, in most applications, multi classifier is required. J. Manikandan et al. [10] designed a SVM multi-classifier with System-on-programmable-chip (SOPC) technology on a Cyclone II FPGA. The accurate and high efficiency onboard HSI multi-classifier satisfying hardware resources and power consumption requirements are still ongoing research concerns.

In this work, a high efficiency multi-classifier using Least Square SVM (LS-SVM) and Hamming-distance judging strategy is proposed and its implementation on a state-of-art Xilinx hybrid chip, ZYNQ, is presented. ZYNQ is a better option for onboard application owing to its low-power consumption and advanced ARM processor with abundant logic resources. To achieve an optimum design solution, an iterative procedure is applied where variant implementation techniques are utilized to produce best possible efficiency. By simultaneously utilizing the control intensive module in processor and computing intensive module in logic cells, a high performance and

^a Corresponding author: maning@hit.edu.cn

low-power consumption computing architecture is designed. The main contributions are:

- Proposing a parallelism multi-classifier based on Hamming-distance judging strategy by combining hardware and software co-design for high performance and energy efficiency to achieve real-time hyperspectral image processing,
- Full experiments performed with real HSI dataset on a ZYNQ platform which is designed for satellite data processing.

2 Mult-Classifier algorithm

According to above discussed features of SVM, it is quite suitable for HSI classification. While SVM is a binary classifier, a multi-classifier is necessary in real processing task which can be implemented by using several binary classifiers in a certain architecture. In order to reduce the requirement of logic resource, LS-SVM is proposed which is the optimum option for onboard application [11].

2.1 Multi-classifier by Hamming-distance decision

To realize a multi-classifier, a *Hamming-distance decision* approach is used to combine the results of several binary classifiers. The result of every binary classifier represents one bit of a code. The bit width of the code is equal to $k(k-1)/2$. For a binary classifier, the outputs are assigned to 1 or 0 to represent different labels. Therefore after the processing of all binary classifiers for a test data, a new result code will be generated. Likewise, for every class label, an identifying code is created when training the sample dataset. The Hamming distance between result code and every identifying code is computed by counting the number of different bits. The class label with minimum distance is then assigned to test data. So for a new result code, Hamming distance (H) between all of the identifying code should be calculated, however not all of the bits in result code are useful for different class labels, a pre-mask should be used for selecting the bits. The Hamming distance is calculated as the following formula:

$$H = \sum_{l=0}^n b_l \quad (1)$$

Where H is the Hamming distance, n is the width of result code and is equal to $k(k-1)/2$, b_l means the l^{th} bit value of T_res which is calculated in (2).

$$T_res = (R_Code \& mask) \oplus I_code \quad (2)$$

Where R_Code is the new result code, $mask$ is the one of the pre-mask which is different for different class label. I_code is the identifying code. The value of identifying code and pre-mask code is dependent on the number of class and the class label of every binary classifier.

In our research we will focus on this approach owing to its higher accuracy than previously described approaches [8, 12].

2.2 SVM binary classifier

An improved version of SVM, LS-SVM is adopted as the binary classifier, because it has assured computing process and needs less logic resources [11].

A LS-SVM classifier should be trained first. The training process can be done in the ground before satellite launching. The on-board processing only involves classification which is described as following:

$$f(x) = \text{sign} \left[\sum_{i=1}^l \alpha_i K(x, x_i) + b \right] \quad (3)$$

where x_i is the training data, $K(x, x_i)$ is kernel function, α are Lagrange multipliers. b is a real constant. α and b can be calculated from training process.

The RBF kernel is the most used kernel function in hyperspectral classification. It is described as:

$$K(x, y) = \exp \left\{ -\|x - y\|_2^2 / \sigma^2 \right\} \quad (4)$$

where σ is the width parameter, and the optimum value of σ in Equation (4) is ascertained by Cross Validation (CV) method together with α and b in training processing.

3 Implementation

To implement such complicated algorithm in heterogeneous platform, several challenges should be overcome. Firstly, a special hardware platform should be designed with considering the computing architecture (including SOPC), logic resource, data cache and data communication interface with the satellite. Secondly, when designing the algorithm, a fully pipeline design should be carefully implemented to get high performance and energy efficiency of the algorithm, the balance point of cycle and resource should be find. The logic part in logic resource and the software part in processor must be working synchronously.

As comprising of processors and special computing elements, a new type of heterogeneous chip, ZYNQ, its lower-power costing and light weight, is used in this research work for onboard classification. The Zynq chip contains two ARM processors (PS partition) and logic resources(PL partition).

Parallelism is implemented to get high performance, and it should be reasonable considering the limitation of hardware resource, data cache size, bus bandwidth, speed of data source and the power supply.

3.1 Implementation of a binary classifier

In this paper, each multi-classifier is made up with a binary LS-SVM classifier and Hamming-distance judging module.

For each binary classifier, the formulas are as described in (3) and (4), Fig.3 shows the architecture of a binary classifier and the architecture of the whole algorithm. For every test pixel which contain several (in

this paper, it is 9) different band data, the binary classifier will run $k(k-1)/2$ times to get every bit of result code.

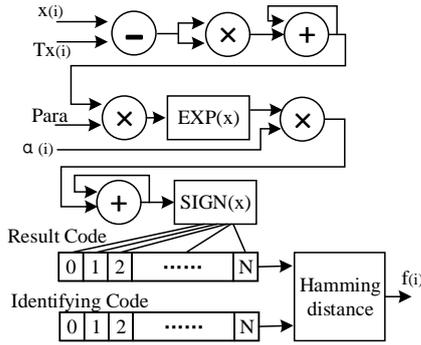


Fig.1. Architecture of. Multi-classifier

According to section II, the function of a 1-vs-1 SVM binary classifier is completed in four steps. In Fig.4, “Bd” stands for spectral band numbers and the “Dim” stands for the training dimension. In the “Two Norm Value” step shown in Fig.4, the value of $\|x - y\|_2^2$ as in Equation (4) is computed, and we need the pixel data and the training pixel data. For one pixel data, it should be processed together with all the training data in above formula, every spectral data should participate the processing which is shown in Fig.3. In the end of this step, there is an iteration which is a challenge for a full pipeline design, all the operations should be finished before starting the second step. Considering that this loop is the inner loop, unrolling can reduce a huge number of processing cycles but not too much extra resources, this step will be implemented in parallel.

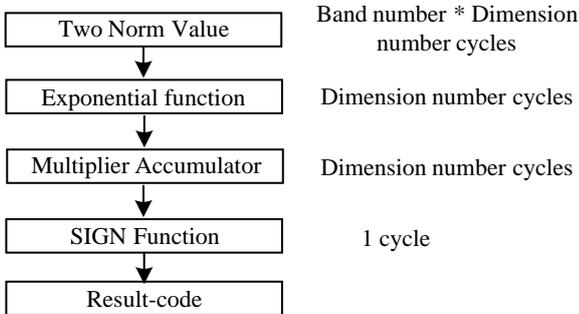


Fig.2. Computing flow of binary classifier

For the step of “Exponential function”, σ is required to calculate the value of Equation (4), and is implemented by using the Exponential IP of Xilinx by using DSP48E and logic resource.

For “Multiplier Accumulator” step, a and b is required. The results of “Exponential function” step are multiplied by a , then accumulating together, this iteration is an outer, unrolling will take a great number of logic resource, and not reduce the processing cycle sharply, this step will be implemented in loop.

Those step can be executed faster by promoting the parallelism at every step but the resource consumption and data communication bandwidth will be sharply increased. So the balance point should be calculated between processing cycles and such factors. For example,

in first step, using 8 couples of subtractor and multiplier, instead of one couple to process 8 training pixels data simultaneously. The cycles of the first step will be reduced to 1/8 compared with raw design. After experiments, we found that a float-point adder required 2 DSP48Es which is the most limited and useful resource in a computing intensive application design, and for subtractor, multiplier and exponential function module the requiring number are 2, 3 and 26 respectively. Considering the total number of DSP48E in our platform is 220, so, when class number is over 4, the device cannot support so much DSP48E for the $k(k-1)/2$ counts of binary classifiers. To realize a scalable design for different number of classes, each multi classifier uses only one unparallelled binary classifier, six multi classifier is designed in our application. For every binary unparallelled classifier, the theoretical total cycle requirement is described in (3). Higher parallelism can be carried out by unrolling the computing loop.

$$Cyc_one = ((Bd + 2) \times Dim + 1) \quad (3)$$

The architecture of the multi classifier by using Hamming-distance is shown in Fig.5.

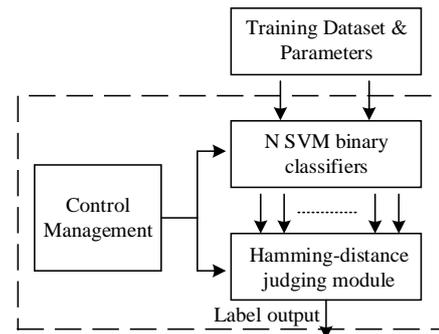


Fig.3. Architecture of Hamming-distance classifier.

It contains three modules: Control Management, SVM binary classifiers and Hamming-distance decision module. Control Management module manages the flow of computing including data input and output, and selects the training data and parameters which decides the final bit location in result code for computing Hamming-distance, those parameters and training data are saved in Training Dataset & Parameters module. There are two approaches to implement SVM binary classifier module, using $k(k-1)/2$ classifiers in parallel to get the results at the same time, this method consumes $k(k-1)/2$ times resources than using only one which required more cycles for processing. Considering the resource limitation of onboard application, so in this paper, one binary classifier is used in each multi classifier.

To find the nearest class, Hamming-distance between result code and identifying code should be computed. By the advantage of logic gates, the result code will be processed in exclusive OR operation with identifying code of every class, then, the result of above operation will be processed in AND operation with a pre-mask code of every class. The number of 1 in every result will be

accounted, the minimum one have the most similar degree with the class label, and set as the data label. Table 1. shows the identifying and pre-mask codes for class number upto six

Table 1. The identifying and Pre-mask codes

Class Number	Identifying Code	Pre-mask Code
Class 1	11111000000000	11111000000000
Class 2	00000111100000	10000111100000
Class 3	000000000111000	010001000111000
Class 4	000000000000110	001000100100110
Class 5	000000000000001	000100010010101
Class 6	000000000000000	000010001001011

3.2 A scalable control program in ARM processor

By just uploading the training dataset and parameters, another new multi-classifier can be launched based on the old classifiers under the control of program. The program which is run on ARM processor controls the flow of computing. It reads HSI dataset and configures the classifier. Program flow is shown in Fig.6.

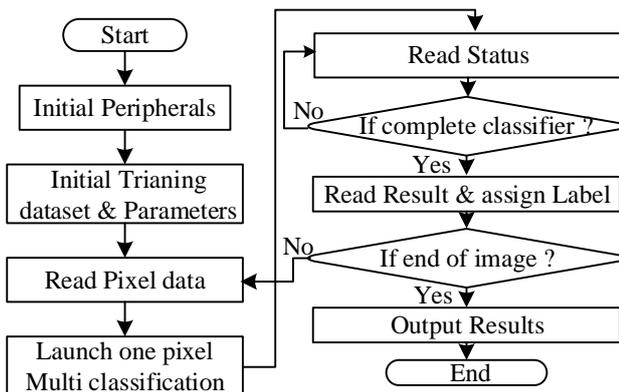


Fig.4. The flow of program in ARM processor.

The program can assure and control the classifier by reading and writing the registers in classifier through AXI Lite interface. In “Initial Peripherals”, the peripherals and the program are prepared for computing, then in “Initial Training dataset & Parameters”, the training dataset and parameter α is sent to Storage module by DMA controller through AXI HP ports, and this can get a high transfer speed and realize a heavy work on processor. In the “Read Pixel data”, the image pixel are transmitted to the binary classifier, the status of the classifier is monitored, after computing the Hamming-distance, multi classifier send the class label of input data to ARM processor. Different classifiers are launched by the program works in parallel. After the processing, results can output through CPCI interface in the demo system. In our experiment, the time and the classifying accuracy will be shown through UART ports on PC.

4 Experiments &Result

To compare efficiency with different platforms, the experiment is carried out on two well-known real HSI datasets. These are produced by Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS) [13], are engaged for testing. AVIRIS can capture 224 bands data for every pixel, from its instrument features, its scan rate is 12Hz, and in every scan, 677 pixels will be produced, the sampling rate is approximately 123.1 μ s/pixel.

To prove our design can realized online real time processing for AVIRIS, the processing time of every pixel is evaluated from the experiments. These two datasets are both provided by the University of the Basque Country [14]. The first image contains 145 \times 145 pixels, and it comes from a mixed agriculture/forestry photo in Northwestern Indiana on June 1992. This image is gathered over the Indian Pines Test Site and shown in Fig.7. The second image is collected from Salinas Valley, California, and its spatial resolution is 3.7meter/pixels comprising of 512 \times 217 pixels.

In our experiments, considering water absorption bands and information redundancy, only 9 spectral bands and 6 classes are used for training and identifying in both datasets. For each class, the number of training and testing pixels is 50 and 100 respectively.

In order to evaluate the performance of our proposed design, four reference designs in different hardware platform are developed in the experiment II. The first one is implemented on HP XW8600 workstation with the configuration of Intel Xeon X5482 having 8 cores at 3.2GHz frequency and 64GB of memory. The same algorithm is implemented in C language on visual studio 2010 development environment. The second reference system is implemented on an embedded system comprising of an ARM cortex-A9 processor at 666.7MHz, Vector Floating Point Unit and 32KB Cache is employed to speed up the computing. The third one is on a state-of-the-art Texas Instruments DSP of TMS320C6778 at 1000 MHz which contain 8 cores to accelerate the processing. The last reference design is on Power PC440 processor, which runs at 400MHz and a FPU in it. The power consumption of these platform are also measured. In order to measure the power consumption more accurately on workstation, the difference of power consumption in running and idle state is used to calculate the power consumption of algorithm. The same 600 pixels data is tested on all the platforms with in the same training dataset, parameters and data precision.

For the Indian Pines Test Site dataset, the time consumption and speedup are shown in Table 2.

Table 2. The speed and power consumption comparing on ZYNQ, ARM, DSP, PowerPC and HP workstation

Platform	ZYNQ	PC	ARM	DSP	PPC
T (μ s/pixel)	25.8	216.3	1321.2	65.7	8516
Power (W)	3.9	103.8	3.3	16.0	9.8
E(mJ/pixel)	0.1	22.4	4.3	1.1	83.5
Speedup	1.0	8.3	51.2	2.54	330
OA (%)	98.3	98.3	98.3	98.3	98.3

In above table, OA means overall accuracy, E stand for energy consumption. From this table, our Hamming-distance multi-classifier on ZYNQ gets 8.3x speedup with about 224x energy saving compared to HP workstation. Compared with ARM, DSP and PPC, our heterogeneous design gain 51.2x, 2.54x, and 330x speedup, and 43x, 11x, and 835x energy saving respectively. By the high frequency clock, embedded FPU, and fast cache, ARM platform gain higher computing speed than DSP and PPC platform. As the lower frequency and the bandwidth limitation between FPU and CPU core, the PPC shows slowest process than other platforms.

Our design gains about 98.3% overall accuracy. With the same training and test data, especially the same classification algorithm, the overall accuracies are the same on different embedded platforms and PC. Under the onboard resources and power limitation, our heterogeneous platform and algorithm architecture is efficient for HSI classification applications.

Comparing with other research on the same dataset, the comparison of overall accuracy is shown in table 3.

Table 3. Comparison with other researches

Methods	OA(%)
Proposed Approach in this paper	98.3
ANN based Adaboost[15]	98.02
Wavelet Networks[16]	82.0
MLRsub[17]	92.5
HA-PSO-SVM[18]	98.2
PGNMF[19]	93.36

From above table, the proposed approach in this paper gain higher accuracy than other research.

For the Salinas Valley dataset, the comparison on power consumption and speedup are shown in Table 4.

Table 4. The speed and power consumption comparison on Zynq, ARM, DSP and HP workstation

Platform	ZYNQ	PC	ARM	DSP	PPC
T (μ s/pixel)	26.5	220.3	1367.2	66.8	8487.4
Power (W)	3.7	106.8	3.1	16.5	9.2
E(mJ/pixel)	0.1	23.5	4.2	1.1	78.1
Speedup	1.0	8.3	51.6	2.6	320.3
OA (%)	97.8	97.8	97.8	97.8	97.8

From above table, for Salinas Valley dataset, the proposed approach can reach 97.8% overall accuracy, and compared to the embedded platform of ARM, DSP and PowerPC, it gains speedup and energy efficiency ranging from 11x ~ 781x.

For online application of AVIRIS, the sampling rate of spectrometer is 123.1 μ s/pixel [20], and our design can realize 27 μ s/pixel classification. So it can fully fill the real-time processing requirements.

5 Conclusions

In this paper, we propose a novel Hamming distance judging strategy based multi-LS-SVM-classifier for HSI classification. By employing parallel logic architecture and flexibility of software in hybrid ZYNQ SOC, we realized the proposed multi classifier with high performance and power efficiency for satellite onboard application. The experiments results on two datasets from AVIRIS demonstrate that the proposed multi-classifier reaches up to 2.5x ~ 330x speed up with 11x ~ 835x energy saving compared to different embedded platforms. At the same time, it gains over 97.8% overall accuracy. So it can realize high overall accuracy and low power consumption real time hyperspectral image classification.

References

1. B. B. Marschallinger, D. Sabel, W. Wagner, "Optimisation of global grids for high-resolution remote sensing data," *Computers & Geosciences*, vol. 72, pp.84-93, November 2014.
2. K. L. Wagstaff, D. R. Thompson and W. K. Abbey, "texture-sensitive instrument classification for in situ rock and layer analysis," *Geophysical Research Letters*, vol. 40, no. 16, pp. 4188-4193, August 2013.
3. S. Lucana, M. Enrico, V. Raffaele, "Highly-Parallel GPU Architecture for Lossy Hyperspectral Image Compression," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (J-STARS)*, vol. 6, no. 2, pp. 670-681, April 2013.
4. S. Bernabe, S. Lopez, A. Plaza, "FPGA Design of an Automatic Target Generation Process for Hyperspectral Image Analysis," in *IEEE 17th International Conference on Parallel and Distributed Systems*, 2011, pp. 1010-1015.
5. G. Camps Valls, L. Bruzzone, "Kernel-based methods for hyperspectral image classification," in *IEEE Transaction on Geoscience and Remote Sensing*, vol. 43, no. 6, pp.1351-1362, 2005.
6. H. Guo, W. Wang, "An active learning-based SVM multi-class classification model," in *Pattern Recognition*, vol.48, no.5, pp. 1577-1597, May 2015.
7. W. H. Chih, J. L. Chih, "A comparison of methods for multiclass support vector machines," in *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415-425, March 2002.
8. S. Hari, F. Agraftoti and D. Hatzinakos, "Design of a Hamming-distance Classifier for ECG-Biometrics," in *Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 3009-3012.
9. A. H. M. Jallad and L. B. Mohammed, "Hardware Support Vector Machine (SVM) for satellite on-board applications," in *Adaptive Hardware and Systems (AHS)*, 2014, pp. 256-261.
10. J. Manikandan, B. Venkataramani, "System-on-programmable -chip implementation of diminishing learning based pattern recognition system," in *International Journal of Machine Learning and Cybernetics*, vol. 4, no. 4, pp. 347-363, Augst 2013.

11. J. Suykens and J. Vandewalle, "Least square support vector machine classifiers," in *Neural Processing Letters*, vol.9, no.3, pp.293-300, March 1999.
12. A. Vempaty, L. R. Varshney, P. K. Varshney, "Reliable Crowdsourcing for Multi-Class Labeling Using Coding Theory," in *IEEE Transactions on Signal Processing*, vol.8, no.4, pp.667-679, AUGUST 2014.
13. R. Green, M. Eastwood, M. Sarture, "Imaging spectroscopy and the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS)," in *Remote Sensing of Environment*, vol.6, no.3, pp. 227-248, September, 1998.
14. Computational Intelligence Group, *Hyperspectral Remote Sensing Scenes datasets*, 2015 [Online]. Available: http://alweb.ehu.es/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes.
15. Q. Sami ul Haq, L. Tao, S. Yang, "Neural Network based Adaboosting Approach for Hyperspectral Data Classification," in *International Conference on Computer Science and Network Technology*, 2011, pp.241-245.
16. P. H. Hsu, H. H. Yang, "Hyperspectral Image Classification Using Wavelet Networks," in *International Conference on Geoscience and Remote Sensing Symposium*, 2007, pp.1767-1770.
17. M. Khodadadzadeh, J. Li, A. Plaza, P. Gamba, "A New Framework for Hyperspectral Image Classification Using Multiple Spectral and Spatial Features," in *International Conference on Geoscience and Remote Sensing Symposium*, 2014, pp.4628-4631.
18. Z. H. Xue, P. J. Du, H. J. Su, "Harmonic Analysis for Hyperspectral Image Classification Integrated With PSO Optimized SVM," in *International Conference on Computer Science and Network Technology*, 2011, pp.241-245.
19. Q. Sami ul Haq, L. Tao, S. Yang, "Neural Network based Adaboosting Approach for Hyperspectral Data Classification," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2131-2146, 2014.
20. Jet Propulsion Laboratory, *AVIRIS Instrument*, 2015 [Online]. Available: <http://aviris.jpl.nasa.gov/aviris/instrument.html>.