Approximate Computing of Remotely Sensed Data: SVM Hyperspectral Image Classification as a Case Study

Yuanfeng Wu, *Member, IEEE*, Xinghua Yang, Antonio Plaza, *Fellow, IEEE*, Fei Qiao, Lianru Gao, *Member, IEEE*, Bing Zhang, *Senior Member, IEEE*, and Yabo Cui

***Abstract*—Onboard processing systems are becoming very important in remote sensing data processing. However, a main problem with specialized hardware architectures used for onboard processing is their high power consumption, which limits their exploitation in earth observation missions. In this paper, a novel strategy for approximate computing is proposed for reducing energy consumption in remotely sensed onboard processing tasks. As a case study, the implementation of support vector machine (SVM) hyperspectral image classification is considered by using the proposed approximate computing framework. Experimental results show that the proposed approximate computing scheme achieves up to 70% power savings in the kernel accumulation computation procedure with negligible degradation of classification accuracy as compared to the traditional ripple carry adder (RCA) precise computation. This is an important achievement to meet the restrictions of onboard processing scenarios.**

***Index Terms*—Approximate computing, energy saving and low power consumption, high-performance computing, hyperspectral remote sensing, onboard processing.**

I. Introduction

REMOTELY sensed images obtained by satellite or spaceborne sensors introduce significant computational requirements. In particular, hyperspectral images comprise hundreds of images at different wavelength channels for the same area on the surface of the earth [1]–[3]. Onboard processing is crucial in some applications requiring fast image analysis response. However, in many cases, onboard processing faces multiple restriction-related processing time, accuracy, and power consumption of the involved hardware [4]–[6]. Traditional high-performance computing architectures, such as field programmable gate arrays (FPGAs) [7], [8], graphic processing units (GPUs) [9]–[11], and multicore processors, have high computing ability and relatively high power consumption, which limits their adoption in real earth observation missions. As a result, trading off performance and energy consumption is an important and emerging line of research [12], [13].

Approximate computing, which aims at translating the precision or output quality into power saving, has recently emerged as a topic of growing interest to achieve low-power-consumption applications. There are several previous researches about approximate computing in digital signal processing, multimedia, and machine learning applications. For instance, an energy-efficient signal processing framework via algorithmic noise tolerance has been developed for low-power digital signal processing [14]. The obtained results show that the input-dependent errors (leading to degradation in algorithmic performance) can be compensated via algorithmic noise tolerance schemes. Here, approximate computing is achieved by considering that all data path computations and bits of stored data in the hardware architecture are not equally significant when shaping the output response of multimedia systems. On the other hand, approximate computing has also been applied to develop a motion estimation technique in the MPEG video encoder, exhibiting the potential for very large power reduction while obtaining reasonable video quality by tolerating errors induced by voltage over scaling and process variations[15]. An error-resilient system architecture applied in Bayesian network probabilistic inference also demonstrated that it is possible to utilize error resilience properties for designing special hardwaresoftware algorithms without incurring into high costs of traditional redundancy-based techniques [16]. In this regard, the concept of scalable hardware design is based on the identification of mechanisms at circuit, architecture, and algorithm levels that can be used to vary the computational effort spent toward the generation of the exact result in the hardware. These mechanisms can be utilized to achieve improved energy efficiency or performance while maintaining acceptable output quality [17]. The most important property for the approximate computing approach in this context is the error tolerance in those applications, in which the data exhibit a large amount of redundancy and a significant decrease on output quality can be disregarded according to human perception.

In this paper, a new approach is proposed for approximate computing of remotely sensed data, which is specifically

Manuscript received October 15, 2015; revised February 03, 2016; accepted February 29, 2016. Date of publication May 03, 2016; date of current version November 30, 2016. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 41301384, Grant 41325004, and Grant 41571349. *(Corresponding author: Lianru Gao.)*

Y. Wu, L. Gao, B. Zhang and Y. Cui are with the Key Laboratory of Digital Earth Science, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100094, China (e-mail: wuyf@radi.ac.cn; gaolr@radi.ac.cn; zb@radi.ac.cn; yabocui@163.com).

X. Yang and F. Qiao are with the Institute of Circuits and Systems, Department of Electronic Engineering, Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: yang-xh11@mails.tsinghua.edu.cn; qiaofei@tsinghua.edu.cn).

A. Plaza is with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura, Cáceres 10071, Spain (e-mail: aplaza@unex.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTARS.2016.2539282

designed for reducing power consumption in onboard hyperspectral image processing applications. The support vector machine (SVM), a well-established classifier for hyperspectral images, is used as a case study for demonstration purposes. First, both the data-level and algorithm-level error resiliences of the SVM technique are analyzed in the context of hyperspectral image classification problems. Our analyses show that the classification procedure exhibits a considerable error tolerance; thus, leaving room for a decrease of output quality that cannot be perceived by humans. In this context, trading off classification accuracy for power savings offers a promising potential. Second, since it can be experimentally observed that accumulate computational kernel dominates the power consumption [18] of the SVM-based computation, circuit-level approximate accumulator implementation techniques are presented. Specifically, the ripple carry adder (RCA) traditionally used for exact computation in hardware implementation is divided into multiple stages, with predictors in the proposed variable latency accumulator. The approximate computation is achieved by removing the error signals of the least significant predictors from the control module. Meanwhile, a hybrid prediction module is used for optimizing the deviation between the approximate and exact accumulator results. The proposed approximate computing schemes are tested using simulated and real hyperspectral image data sets. Experimental results show that the proposed approximate computing scheme achieves up to 70% power savings (by voltage over scaling technique) with negligible degradation of classification accuracy. This is an important achievement to meet the restrictions of onboard processing scenarios.

This paper is structured as follows. In Section II, both the data-level and algorithm-level error resiliences of the SVM classifier are analyzed in a hyperspectral image classification application. In Section III, approximate computing techniques for reducing power consumption are presented in the considered case study. In Section IV, simulated and real hyperspectral data sets are used to evaluate the classification accuracy and power saving percentage of the proposed approximate computing approach. Section V concludes this paper with some remarks and hints at plausible future research lines.

II. Analysis of Error Resilience in Hyperspectral Image Classification

In order to empirically illustrate the algorithmic resilience in the context, we consider hyperspectral image classification as a case study. Specifically, we use the SVM, which is a popular kernel-based supervised algorithm for hyperspectral image classification. First, the SVM algorithm is briefly introduced, and then the error resilience of the classification results using a real hyperspectral image data set is analyzed.

A. Hyperspectral Image Classification With SVMs

Hyperspectral classification with SVMs includes two main procedures: training and prediction [19], [20]. In the training procedure, each pixel in a training set is assigned with a class label. The training algorithm tries to find the optimal separating hyperplane that maximizes the margin between the closest pixels. The boundary pixels, which are called *support vectors*, are used to create a decision surface. In the prediction procedure, each unlabeled pixel is assigned with a label based on the relative distance to the hyperplane. The decision function is given by

$$Label(x) = \text{sign}\left(\sum_{i=1}^{N} a_i * y_i * K(x_i, x) + b\right) \quad (1)$$

where x is the input pixel vector; x_i is the ith support vector; a_i and y_i are the ith Lagrange coefficient and the corresponding classification label, respectively; N is the number of support vectors; and b is the decision offset coefficient. Considering a specific hyperplane, the Lagrange coefficient, the support vectors, and the decision offset coefficient are fixed. K is a kernel function used to transform the original data to a specific feature space [21]. In the experiments, a nonlinear SVM classifier based on the Gaussian radial basis kernel function (RBF) is considered. The RBF can be defined by

$$K(x_i, x) = \exp(-\gamma * \|x_i - x\|^2) \quad (2)$$

where γ is the RBF coefficient, determined after scanning the image data, x is the input pixel vector, and x_i is the ith support vector.

B. Hyperspectral Image Data Sets

The image scene used for experiments in this work was collected by the airborne visible-infrared imaging spectrometer (AVIRIS) instrument, which was flown by the National Aeronautics and Space Administration (NASA)'s Jet Propulsion Laboratory (JPL). This scene, with a size of 145 lines by 145 samples, was acquired over a mixed agricultural/forest area. The scene comprises 220 spectral channels in the wavelength range from 400 to 2500 nm, nominal spectral resolution of 10 nm, moderate spatial resolution of 20 m by pixel, and 16-bit radiometric resolution. After an initial screening, several spectral bands were removed from the data set due to noise and water absorption phenomena, leaving a total of 200 radiance channels to be used in the experiments.

Fig. 1(a) shows a false color composition of the AVIRIS Indian Pines scene; Fig. 1(b) shows the ground-truth map available for the scene, displayed in the form of a class assignment for each labeled pixel, with 16 mutually exclusive ground-truth classes (in total: 10 366 samples). Table I shows the number of samples that will be used for classification purposes.

C. Data-Level Error Resilience Analysis

In order to analyze the data-level error resilience, noise (random error data) is injected in the original hyperspectral image to evaluate the impact on the overall classification accuracy. The AVIRIS Indian Pines scene is used in the experiment. It has 10 366 labeled samples (see Table I). Among these labeled data, 1036 samples are used for training and the rest (9330 samples) are used for testing. Thus, there are $9330 \times$

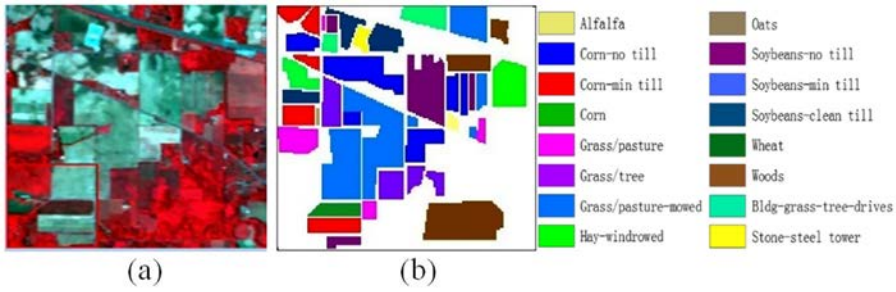


Fig. 1. (a) False color composition of the AVIRIS Indian Pines scene. (b) Ground-truth map available for the scene.

TABLE I
LABELED SAMPLES OF EACH CLASS IN THE AVIRIS INDIAN PINES DATA

Class	Name	Number of samples
CA-1	Alfalfa	54
CA-2	Corn-no till	1434
CA-3	Corn-min till	834
CA-4	Corn	234
CA-5	Grass/pasture	497
CA-6	Grass/tree	747
CA-7	Grass/pasture-mowed	26
CA-8	Hay-windrowed	489
CA-9	Oats	20
CA-10	Soybeans-no till	968
CA-11	Soybeans-min till	2468
CA-12	Soybeans-clean till	614
CA-13	Wheat	212
CA-14	Woods	129
CA-15	Bldg-grass-tree-drives	380
CA-16	Stone-steel towers	95

200 (about 1.87×10^6) floating-point numbers involved in the classification calculation.

As shown in Fig. 2, noise is added to the least significant bits (LSBs) of the original hyperspectral data. The LSBs are extended from low to high in binary format, which formed a series of new "corrupted" test data. It should be noted that the noise is only added in pixels to be labeled in the prediction procedure. Thus, the support vectors obtained from the training procedure maintain the same characteristics to make the classification results comparable. The simulation is performed using the $libsvm$ library and $MATLAB$ software; the $svmtrain$ function parameters are set to be consistent throughout the experiments, in which the kernel function variable r is set to 3.92 and C is set to 69.60 (obtained by genetic algorithm).

The detailed data-level error resilience analysis of the SVM-based hyperspectral image classification with AVIRIS Indian Pines scene includes the following steps.

1) Normalize the original data and save it as 64-bit floating-point data.
2) Randomly select a certain number of pixels.
3) Convert the selected pixel values from decimal into binary system.
4) Generate a random number and convert into binary system.
5) Inject the random number in the LSBs of the selected pixels. It should be noted that, for each pixel, the random number is different.
6) Convert the new data from binary into decimal system.

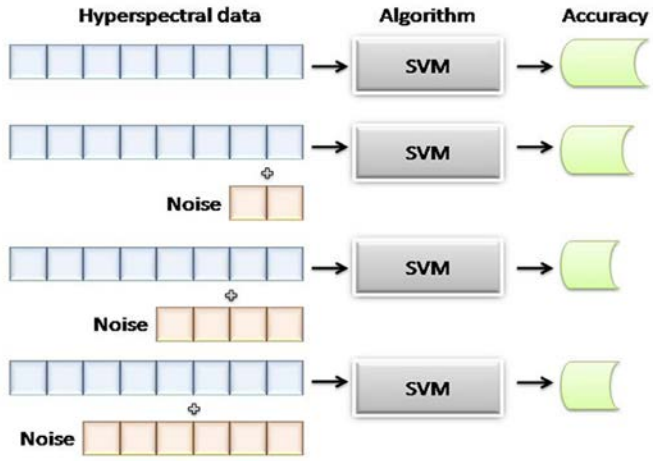


Fig. 2. Data-level error resilience analysis of the SVM hyperspectral image classification process.

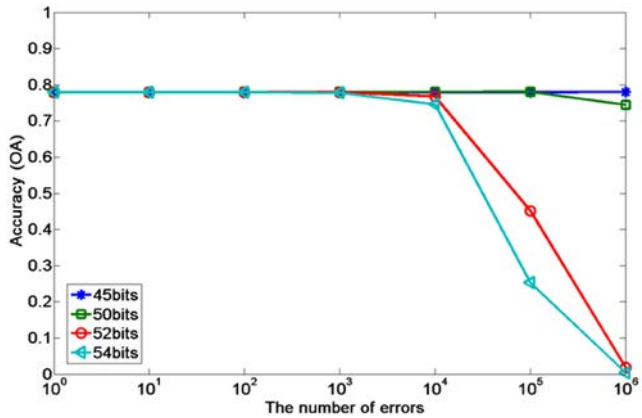


Fig. 3. Accuracy of the SVM classification process for the AVIRIS Indian Pines data set after injecting different ranges of noise into the LSBs.

7) Evaluate the overall classification result by using the new "corrupted" data.

For illustrative purposes, the number of selected pixels will be increased by a factor of 10 each time. For a certain number, the range of the LSBs will be varied. The results are presented in Fig. 3, in which the x-axis represents the number of errors injected, and the y-axis represents the overall classification accuracy (different curves correspond to different ranges of LSBs).

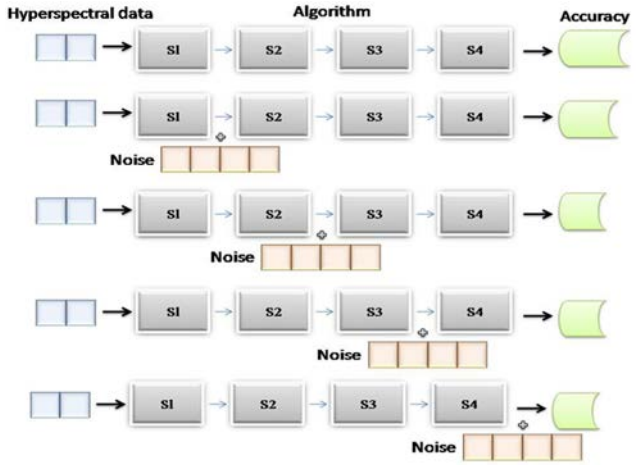


Fig. 4. Algorithm-level error resilience analysis of SVM-based hyperspectral image classification.

From Fig. 3, it can be seen that, when errors are injected in the 45 LSBs (even with a probability of 100%), they do not have any impact on the final classification accuracy. Similarly, injecting errors in the 50 LSBs has negligible impact on classification accuracy. Moreover, injecting errors with 10^4 numbers in 52 of the 64 bits have negligible impact on classification accuracy.

D. Algorithm-Level Error Resilience Analysis

In order to analyze algorithm level error resilience, Gaussian noise is injected into the intermediate results of the classification algorithm step by step, and we further evaluate their impact on the final classification accuracy (as shown in Fig. 4). As in the data-level error resilience analysis experiment, the noise is only added in the prediction procedure and the support vectors obtained from the training procedure maintain the same properties to make the classification results comparable.

The input data are the original AVIRIS Indian Pines scene, and the properties of the Gaussian noise function are set as follows: the mean value is set to zero and the variance value varied from zero to one to simulate the injected error magnitude. The simulation environment and parameters of the function are consistent with the data-level error resilient analysis experiment.

According to the properties of the SVM classification algorithm, the prediction procedure can be divided into four modules.

1) *Kernel computation:* perform the dot product between a test vector and the support vector.
2) *Coefficient multiplication:* perform Lagrange coefficient and classification label multiplication with the kernel computation result.
3) *Accumulation:* add the coefficient multiplication results from the first support vector to the Nth support vector.
4) *Offset summation:* add the decision offset coefficient to the accumulation result.

To make it clear, each time Gaussian noise is only added into one module. The obtained results are presented in Fig. 5.

From Fig. 5, it can be seen that, when the variance of the Gaussian noise injected in the kernel computation module or in the coefficient computation module is less than 0.02, there is no impact on the final classification accuracy. Similarly, when the variance of the Gaussian noise injected in the accumulation module or in the offset summation module is less than 0.2, there is no impact on the final classification accuracy. It should be noted that the kernel computation and coefficient computation modules entirely consist of dot product and multiplication computations. On the other hand, the accumulation and offset summation modules entirely consist of summation computations. Dot products and multiplications are more sensitive to data errors, while the summation operation is not as sensitive to data errors. As a result, dot products and multiplications enlarged the data errors much faster than summations.

After the conducted analysis of data-level and algorithm-level error resilience, results suggest that design techniques that trade off the accuracy of the computations for energy saving are worth being further investigated. Real hyperspectral images exhibit a large amount of redundancy, and this may allow us to decrease output quality due to the limited human perception. The error resilience properties are the most important features when evaluating the feasibility of an approximate computation approach. Since the kernel accumulation computation module is the most computationally intensive of the SVM classification process (as shown in reference [18], it spends 85%–95% of the computation time), this increases the potential to exploit approximate computation in this context.

III. Circuit Design of an Approximate Accumulator for Low-Power Computing

After experimentally assessing that SVM-based hyperspectral image classification exhibits both data-level and algorithm-level error resilience, in this section, it is illustrated that there is no need to follow a traditional (strict) design methodology in which the computing procedure is mapped exactly into a hardware implementation in this particular case. Thus, this property could be exploited to achieve more power efficiency when the computing procedure is mapped to an approximate hardware implementation [22]. As pointed out in [23], approximate computing technology aims at trading off the precision or output quality of the algorithm into power savings and has been presented as a promising method to further increase the power efficiency of hardware implementations. As shown in [18], it can be experimentally observed that accumulate computational kernel dominates the power consumption of the SVM-based computation. Considering the error resilience property, in this section, circuit-level approximate design techniques are applied to implement the kernel computation operation in hardware.

A. Variable Latency Accumulator

The traditional RCA commonly used for exact computation is divided into different stages with predictors in the variable latency accumulator [24], [25]. The advantage of using a variable latency accumulator is that the long critical path is divided into several shorter paths. Thus, the proposed approach can improve the clock frequency or maintain the same clock frequency by a lower voltage supply. As a result, trading-off

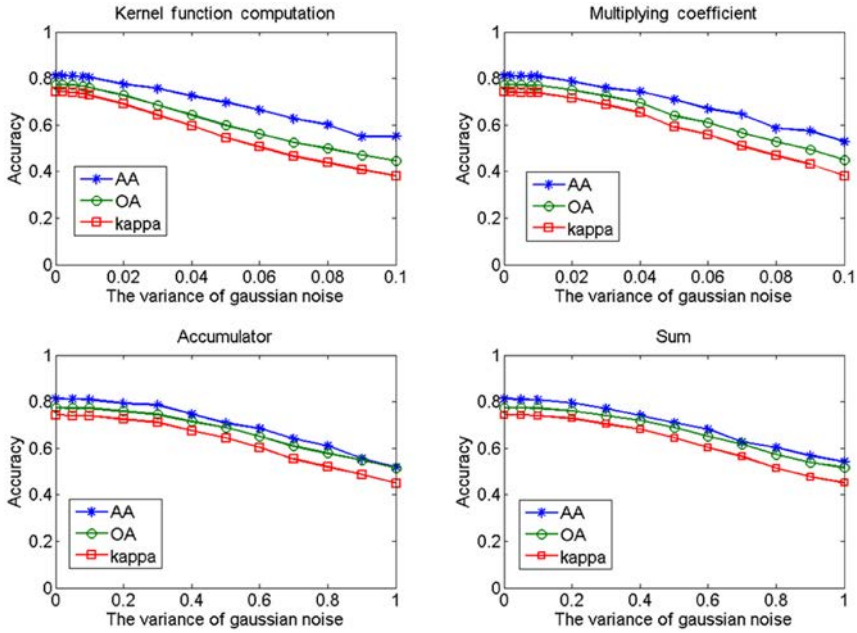


Fig. 5. OA of the SVM classification process for the AVIRIS Indian Pines data set after injecting different levels of noise into the intermediate modules of the classifier.

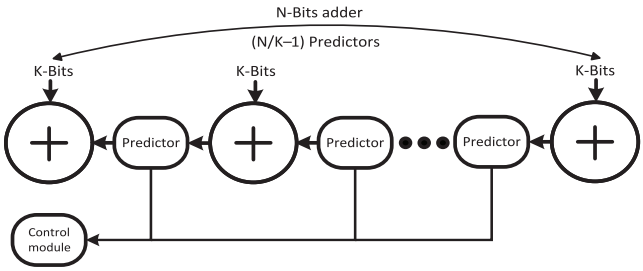


Fig. 6. Variable latency accumulator.

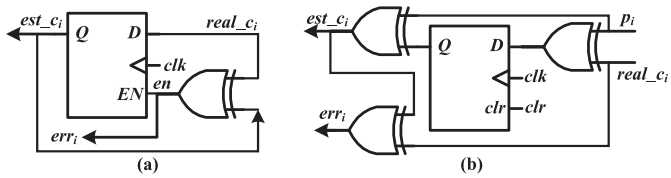


Fig. 7. (a) 1-bit reserve-forward predictor. (b) 1-bit synchronous predictor.

performance for power consumption can be achieved using a voltage scaling technique. That is the theoretical basis of the variable latency accumulator for low power consumption. It should be noted that, in theory, using multiple stages with predictors may lead to multiple clock cycles consumption. Thus, decreasing the average clock cycles for one operation is significant in practical computation.

However, this probability is very small due to the strong data dependence in real implementations in which, in most cases, one cycle is enough for the computation if the adder is applied to the less significant parts of the algorithm and the supply voltage can decrease to a lower level due to the shorter path that is obtained after prediction.

As shown in Fig. 6, the N-bits RCA is divided into different fragments comprising K bits for each stage, and N/K-1 predictors are inserted into the adder. If a wrong prediction happens at the end of each cycle, the predictor will produce an error signal that emerges into a control module. The control module will hold the input-output D-Flip-Flops so that another clock cycle is allocated to the adder to correct the wrong prediction until no error signal exists.

According to the descriptions in [26], the critical path of the accumulator can be broken through the prediction of the carry signals, in which all the prediction methods can be summarized into two categories as shown in Fig. 7. Suppose the input data are $A_{1...n}, B_{1...n}$ and k bits are distributed to each stage.

In Fig. 7(a), for i-stage$(i > 0)$, the estimation carry (est_ci) is the sampling value of the carry signal of last operation preserved by the D-Flip-Flop and further pushed into the xor gate to compare with the real carry $(real\_ci)$ signal so that the err_i can be generated and emerged into the control module in Fig. 6. In this design, all the err_i signals emerge through the or gate in the control module. When the predictor is inserted into the accumulator, the est_ci will be $real\_ci$ in last operation at first cycles if en signal is high. At the end of the first cycle, the en signal will be high if the est_ci is not equal to the current $real\_ci$. At this time, the err_i will also be high and the D-Flip-Flop for the input and output of the whole addition will be latched and another clock cycle will be used so that the $real\_ci$ signal will be forwarded.

In Fig. 7(b), the prediction process is different. The predictions will be carried out when new data are pushed and the p_i signal is the combination logic result of current input data $A_{1...n}$ and $B_{1...n}$, as $p_i = A_j B_j$, $j = ik$. The output of the D-Flip-Flop in Fig. 7(b) will be cleared to "0" at first cycle so that the p_i signal can be forwarded through the xor gate to est_ci, which is then compared with $real\_ci$ signal to generate err_i. The clr signal (clear signal) will be produced by the control module in Fig. 6 and keep invalid until no misprediction incurs.

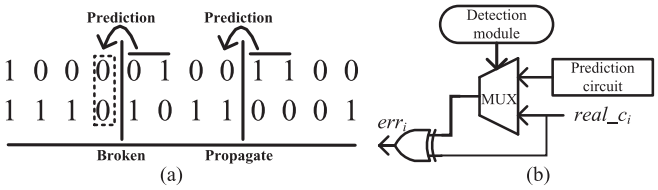


Fig. 8. Illustration of the process adopted for removing redundant cycles. (a) Data illustration for redundant cycles. (b) Circuit scheme for detection.

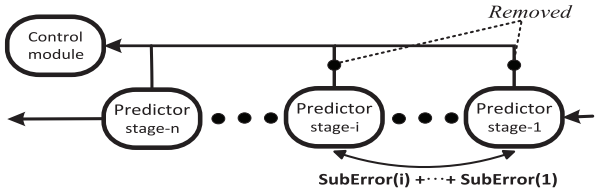


Fig. 9. Approximate variable latency accumulator.

In order to decrease the average clock cycles for one computation, a module to detect the pattern of input data is inserted into the predictor to further improve the performance of the variable latency accumulator. As shown in Fig. 8(a), the prediction is redundant if the carry signal path has already been broken with the current input data. Thus, the predictor can be modified to detect this input pattern. As shown in Fig. 8(b), the detection module can be realized as $det_i = A_{j+1} \oplus B_{j+1}$ or $det_i = (A_{j+1} \oplus B_{j+1}) \cap (A_{j+2} \oplus B_{j+2}), j = ik$. When the critical path is detected to be broken naturally with the input data, the prediction process described previously will be ignored. With the detection module, the average clock cycles will be reduced and the whole performance of the accumulator will be improved.

B. Approximate Variable Latency Accumulator

An approximate variable latency accumulator is implemented, which can be easily extended to an approximate computing structure by removing the error signals produced by the least significant predictors from the control module. This means that no extra clock cycles will be allocated, even if there are wrong predictions in these least significant predictors. With this, the clock cycle consumption of the accumulator will be further reduced, thus trading off for power consumption.

As shown in Fig. 9, the error signals produced by the least significant predictors are directly removed from the control module. In this way, the corresponding errors in different stages will be generated, and the clock cycles for correcting the wrong predictions can be further reduced. The ith stage computation error, err_i, is the sum of all the suberrors which can be calculated by

$$err_i = \sum_{m=1}^{i} SubError(m). \tag{3}$$

The magnitude of each $SubError$ is decided by the position of the corresponding predictor. In other words, the weight of the $SubError$ is different and increases with the second order of the magnitude. A large error will be generated if more err_i signals are removed and, consequently, the cycle consumption will inversely decrease. This means that the approximate variable latency accumulator enables trading-off output quality (computation accuracy) and performance. It should be noted that the number of err_i signals removed from the least predictors is not fixed. More err_i signals are removed means more error will be introduced and less clock cycles will be consumed.

However, large amount of errors may be introduced with this approximate scheme. Let us use an example the 1-bit synchronous predictor in Fig. 7, considering the scheme of $p_i = A_j B_j$, $j = ik$. When A_j and B_j are equal (both are logic "1" or logic "0"), the p_i value will be same with the real carry value. When A_j and B_j are different (also called propagate state in adder design), the p_i value will be logic "0", which led to misprediction if the real carry signal is logic "1". Thus, a negative error will be generated, and the accumulate operation will amplify the error after hundreds of calculations. For the 1-bit reserve-forward prediction scheme, as p_i is always the sampling value reserved in last addition operation, the error introduced from each predictor in every addition has positive and negative deviation. This opposite error direction can produce offset in practical accumulation, which lets the approximate value not deviate largely from the real value. It should be noted that the performance of accumulator with the reserve-forward prediction is much lower than with the synchronous prediction.

C. Prediction Improvement With Error Compensation

In order to control the error magnitude, an approximate variable latency accumulator with hybrid prediction is used for optimizing the deviation. For the precise computation part, the circuit scheme should be realized as 1-bit synchronous predictor $p_i = A_j B_j$ to keep a high-speed processing when the voltage is scaled down. The 2-bit detection module $det_i = (A_{j+1} \oplus B_{j+1}) \cap (A_{j+2} \oplus B_{j+2})$, $j = ik$ should also be adopted to the predictor, which means that the correct part in the accumulator is equipped with the highest performance. For the approximate computation part, in order to avoid negative error value and produce offset in accumulation, the prediction circuit should be realized as 1-bit reserve forward scheme in this approximate when the err_i signal is removed from the control module.

As shown in Fig. 10, when the err_i signals are removed from the predictors to translate certain accuracy into performance improvement or power savings, the corresponding prediction circuit parts are replaced with 1-bit reserve-forward implementation. Thus, in the hybrid prediction scheme, the least significant predictors (err-signals removed) employ reserved prediction while the significant predictors (with err-signals) employ synchronous prediction. The reserved prediction makes the error generated each time have different direction, from which the error can be offset in a long series of calculations. The final deviation between the approximate result and the exact result will be very small. The clr represents the clear signal which used to trigger clear operation.

The hybrid prediction scheme can be considered as an optimization to the approximate variable latency accumulator in trading off computation accuracy for power consumption. Thus, the hybrid prediction scheme enables the whole approximate

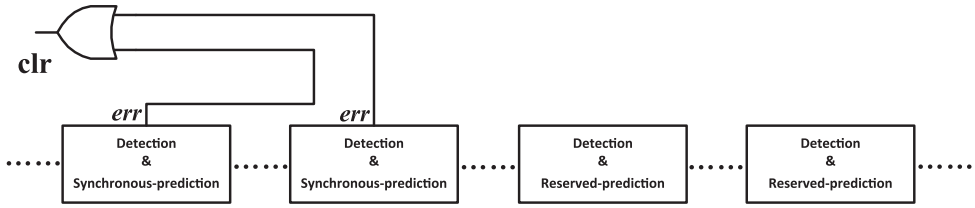


Fig. 10. Approximate variable latency accumulator with improved prediction method.

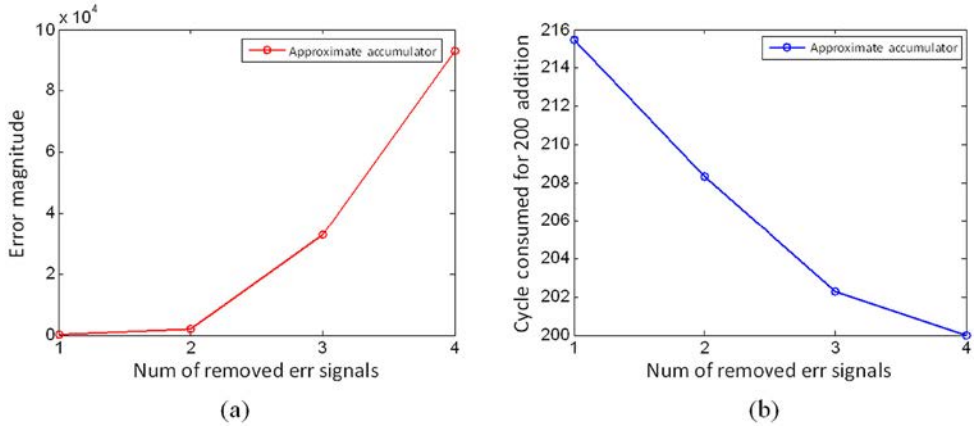


Fig. 11. (a) Error magnitude versus the number of removed error signals. (b) Cycle consumption versus the number of removed error signals.

accumulator to achieve high performance while keeping a small error magnitude.

IV. Experimental Results

In this section, experimental evaluations of the approximate computing implementation are provided. For classification accuracy and power saving comparisons, the traditional RCA and the approximate accumulators are implemented in $Verilog$ under $SMIC$ 65 nm process and then synthesized using the *DesignCompiler of Synopsys*.

The power consumption is measured by using the $PrimeTime$ tool. In the following, different schemes of adders used in this work are given below.

1) Traditional RCA (a precise computing implementation based on the traditional RCA, used as a benchmark for comparisons).
2) Approximate accumulators (approximate computing implementation based on the proposed approximate variable latency accumulator with improved prediction method, the number of removed error signals is increased from zero to four).

A. Performance Evaluation With Simulated Data

First, the performance of newly proposed approximate computation schemes are tested using simulated data. The test operation is the vector dot product. x_i and y_i are defined as vectors of dimensionality 200×1. app_i is the approximate dot product result, and $correct_i$ is the traditional correct dot product result. The error magnitude between the approximate computation and the exact computation can be calculated by the following expression:

$$ErrorMagnitude = \sqrt{\frac{\sum_{i=1}^{N}(app_i - correct_i)^2}{N}}. \quad (4)$$

Meanwhile, 10^6 times of dot products were simulated in the experiment. The error magnitude and corresponding cycle consumption were recorded with an increase in the number of removed err signals (see Fig. 11). It can be seen that, when the number of err signals is increased, the error magnitude increases and the cycle consumption decreases. It should be noted that there are 200 additions for each accumulation and performed 10^6 times. According to the data-level and algorithm-level error resilience presented in Section II, the deviation between the approximate computing value and the precise computing value will be quite small, which especially applies to real hyperspectral image data sets.

B. Classification Accuracy Evaluation With Real Hyperspectral Images

The classification accuracy can be evaluated by measuring the average accuracy (AA), overall accuracy (OA), kappa statistical coefficient (kappa), and individual class accuracy (CA) [27]–[29]. It should be noted that, when the approximate accumulator removed zero error signals, it achieves the same classification accuracy with the traditional RCA implementation. Therefore, only the change in classification accuracy with different number of removed error signals of the approximate accumulator will be counted.

1) Results With AVIRIS Indian Pines Data Set: Fig. 12 shows the obtained classification maps for the AVIRIS Indian Pines scene based on the following implementations: (a) result obtained by removing zero error signals; (b) result obtained by

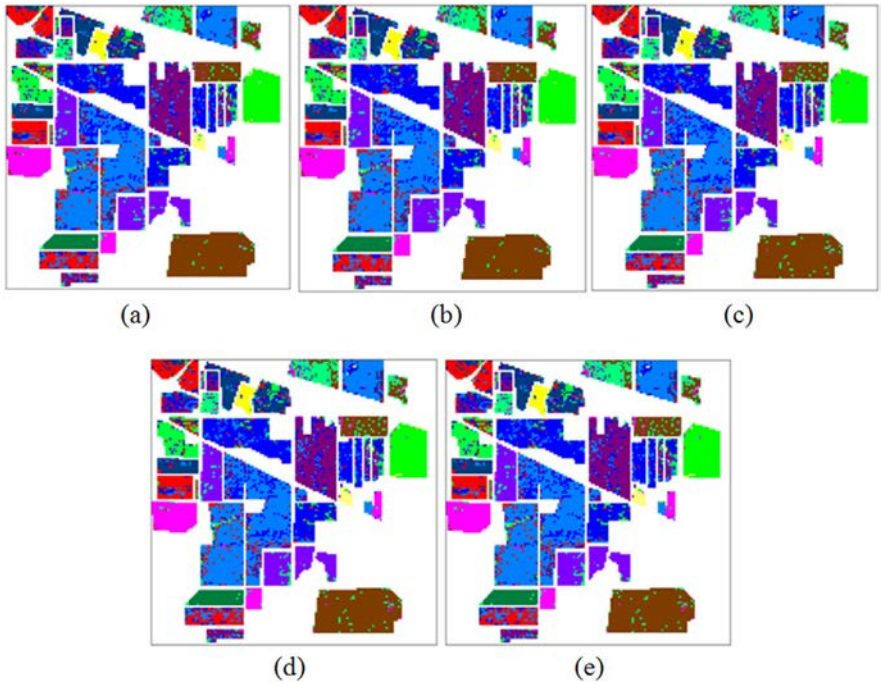


Fig. 12. Classification maps obtained for the AVIRIS Indian Pines data set by using different configurations of approximate accumulators. (a) Zero removed error signal. (b) One removed error signal. (c) Two removed error signals. (d) Three removed error signals. (e) Four removed error signals.

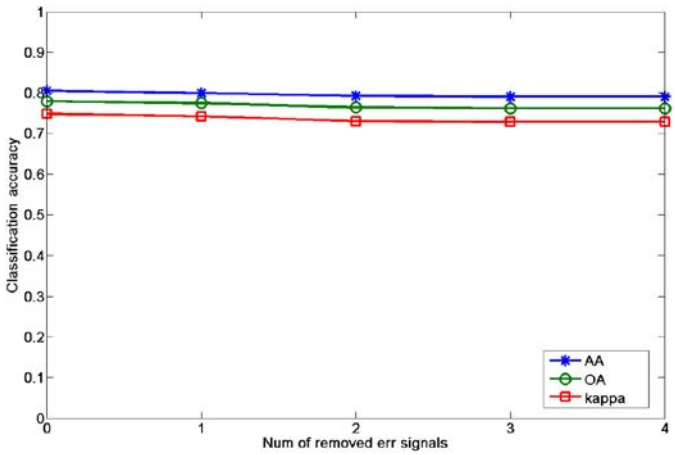


Fig. 13. AA, OA, and kappa accuracy of SVM classification for AVIRIS Indian Pines data set by using different configurations of approximate accumulators.

removing one error signals; (c) result obtained by removing two error signals; (d) result obtained by removing three error signals; and (e) result obtained by removing four error signals. It can be seen that all the implementations result in almost the same final classification maps (to the point that, from a human vision perspective, their differences can be neglected).

In order to quantitatively analyze the approximate computation schemes, the AA, OA, and Kappa coefficients are plotted in one map by increasing the number of removed error signals (see Fig. 13).

From Fig. 13, it can be concluded that the original AA is about 0.805 with no error signal removed. The second AA is about 0.799 with one error signal removed. The third AA is about 0.792 with two error signals removed. The fourth AA is 0.790 with three error signals removed. The fifth AA is 0.790 with four error signals removed. It can be observed that similar changes occur in the curves of OA and Kappa. Most importantly, the output quality of the algorithm can stay in an acceptable level when increasing the number of removed error signals.

To further quantitatively analyze the approximate computation schemes, the individual classification accuracy of the 16 classes contained in the AVIRIS Indian Pines ground truth are presented in Table II. With the number of removed error signals increasing, classes CA-2, CA-4, CA-5, CA-7, CA-8, CA-9,and CA-11 (see Table I) are consistent with the original accuracy, while the others exhibit a slight decrease.

2) Results With ROSIS University of Pavia Data Set: The ROSIS University of Pavia data set was acquired by the reflective optics system imaging spectrometer system (ROSIS) optical sensor, which was operated by the Deutschen Zentrum for Luftund Raumfahrt (DLR, the German Aerospace Agency) in the framework of the HySens project, managed, and sponsored by the European Union. The image has a size of 610×340 pixels, with very high spatial resolution of 1.3 m/pixel. The number of data channels in the acquired image is 103 (with spectral range from 430 to 860 nm). Fig. 14(a) shows a false color composite of the ROSIS University of Pavia scene; Fig. 14(b) shows nine reference classes of interest, which comprise urban features, as well as soil and vegetation features. The nine classes of training samples are randomly selected from the reference map.

The classification maps by different configurations of approximate accumulators are presented at the bottom of Fig. 14; Fig. 14(c) shows the result obtained by removing zero error signals; Fig. 14(d) shows the result obtained by removing one error signals; Fig. 14(e) shows the result obtained by removing two error signals; Fig. 14(f) shows the result obtained

TABLE II
INDIVIDUAL CLASS ACCURACIES AFTER APPLYING THE SVM TO THE AVIRIS INDIAN PINES DATA SET BY USING DIFFERENT CONFIGURATIONS OF APPROXIMATE ACCUMULATORS

Class	Individual CA				
	0	1	2	3	4
Alfalfa	85.19	81.48	77.78	77.78	77.78
Corn-no till	69.42	69.05	69.05	69.20	69.20
Corn-min till	66.84	66.45	63.14	64.33	64.33
Corn	80.47	79.29	81.07	81.66	81.66
Grass/pasture	90.85	90.85	91.78	92.25	92.25
Grass/tree	92.70	92.25	89.87	89.42	89.42
Grass/pasture-mowed	76.92	76.92	76.92	76.92	76.92
Hay-windrowed	96.45	96.45	96.22	95.98	95.98
Oats	70.00	70.00	70.00	70.00	70.00
Soybeans-no till	74.89	74.32	72.18	70.72	70.72
Soybeans-min till	72.75	72.20	71.56	71.86	71.86
Soybeans-clean till	68.14	66.85	67.96	66.85	66.85
Wheat	95.92	95.92	94.56	94.56	94.56
Woods	93.05	92.89	88.67	88.09	88.09
Bldg-grass-tree-drives	61.17	59.87	66.67	64.40	64.40
Stone-steel towers	93.55	93.55	90.32	90.32	90.32

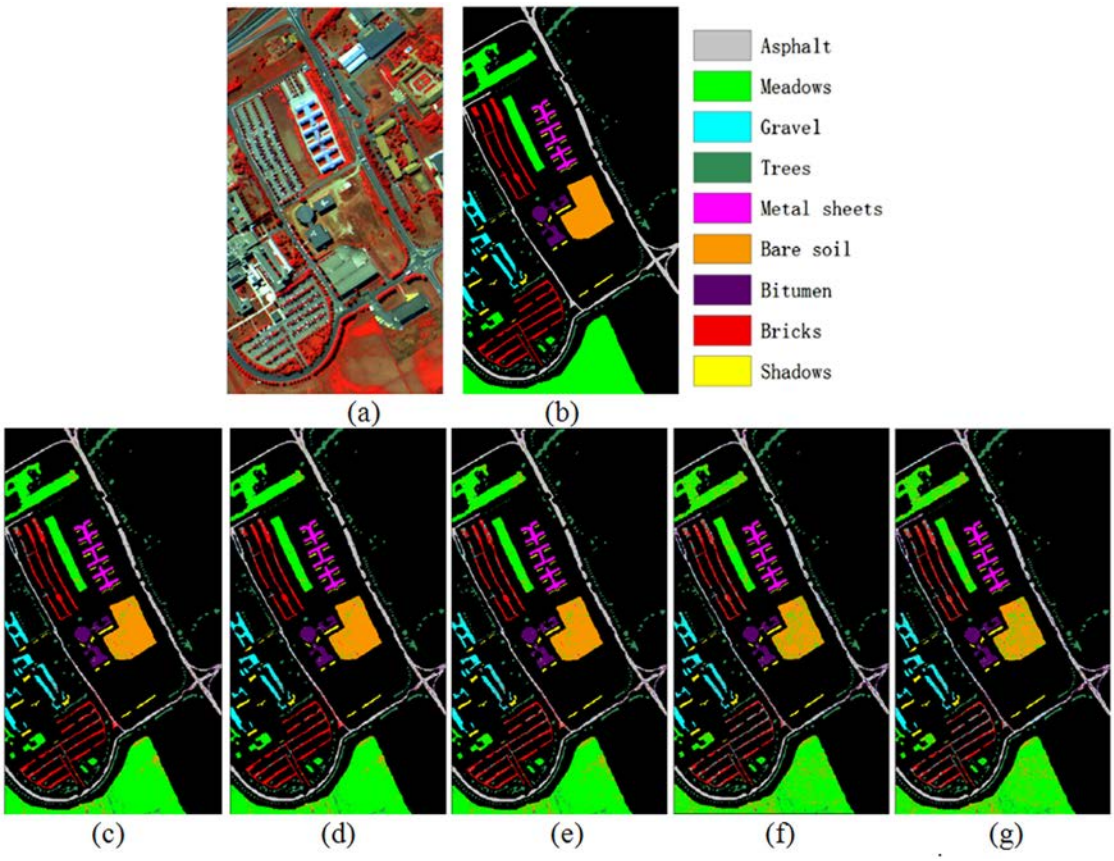


Fig. 14. Classification maps obtained for the ROSIS University of Pavia data set using different configurations of approximate accumulators. (a) False color composition of the image. (b) Reference map. (c) Zero removed error signal. (d) One removed error signal. (e) Two removed error signals. (f) Three removed error signals. (g) Four removed error signals.

by removing three error signals; Fig. 14(g) shows the result obtained by removing four error signals. It can be seen that the maps from Fig. 14(c) to (e) have almost the same classification results; the maps Fig. 14(f) and (g) have a small amount of noise in the bare soil and meadows classes as compared with the first three maps. These noise effects may be acceptable from a human vision perspective in some application scenarios.

In order to quantitatively analyze the approximate computing framework with the ROSIS Pavia University data set, the AA, OA, and Kappa coefficients are plotted in one map by increasing the number of removed error signals (see Fig. 15). From this figure, it can be seen that the original kappa is about 0.899 with no error signal removed. The second kappa is about 0.898 with one error signal removed. The third kappa is about 0.859 with

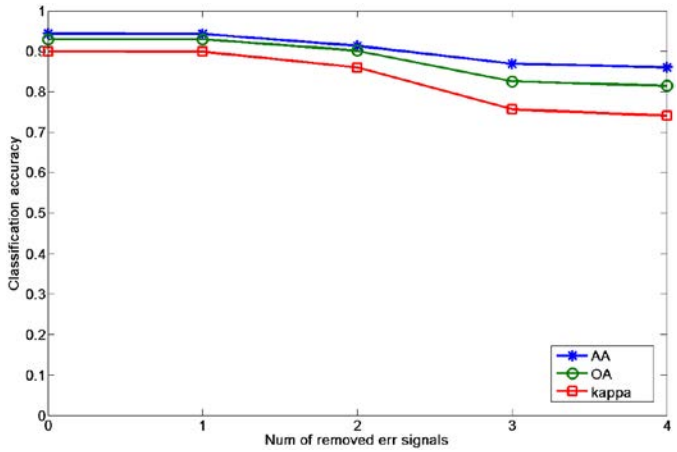


Fig. 15. AA, OA, and kappa accuracy of SVM classification for ROSIS University of Pavia scene by using different configurations of approximate accumulators.

TABLE III
INDIVIDUAL CLASS ACCURACIES OF SVM CLASSIFICATION FOR ROSIS UNIVERSITY OF PAVIA DATA SET BY USING DIFFERENT CONFIGURATIONS OF APPROXIMATE ACCUMULATORS

Class	Individual CA				
	0	1	2	3	4
Asphalt	89.12	89.03	87.79	78.36	78.08
Meadows	93.54	93.55	90.56	83.96	82.68
Gravel	88.36	88.19	80.84	80.76	78.04
Trees	98.53	98.53	98.16	95.17	94.43
Metal sheets	100	100	100	100	100
Bare soil	95.24	95.27	94.03	82.85	80.94
Bitumen	94.57	94.57	87.56	92.76	92.76
Bricks	89.69	89.73	82.68	68.04	67.07
Shadows	100	100	100	100	100

two error signals removed. The fourth kappa is 0.756 with three error signals removed. The fifth AA is 0.740 with four error signals removed. Similar changes can be observed in the curves of OA and AA. On the whole, the output quality of the algorithm can stay in an acceptable level when increasing the number of removed error signals.

To further quantitatively analyze the approximate computing framework with the ROSIS University of Pavia data set, the individual classification accuracy of the nine classes contained in the reference map is presented in Table III. When the number of removed error signals is less than three, individual class values are almost the same when compared to the precise computation results. If we continue to increase the number of removed error signals, there is an obvious decrease in the accuracy for the bare soil, meadows, and bricks classes.

From the classification maps and quantitative analysis of the AVIRIS Indian Pines and the ROSIS University of Pavia data sets, it can be concluded that the approximate computing implementations achieve manageable decrease on output quality and comparable classification accuracies with regard to the precise computing results. It should be noted that the number of error signals removed from the least predictors is not fixed; actually, it depends on the practical application. When more error signals are removed, that means bigger deviation will be introduced between precise computation and approximate computation. Correspondingly, less clock cycles will be consumed and higher power saving percentage will be achieved.

C. Power Saving Evaluation With Real Hyperspectral Images

In this section, the power consumption and computing efficiency of the proposed approximate computing implementations are quantitatively analyzed. The corresponding results of traditional RCA for precise computing implementation are used as a benchmark for comparisons. The static timing analysis is performed by using the $SynopsysPrimeTime$ tool. The unit delay and clock cycles of these implementations can be directly measured. Then average delay equals the unit delay multiplied by the clock cycle. The computing efficiency (speedup) can be obtained by ratio of average delays between traditional RCA and approximate accumulators. The power consumption can be calculated through voltage scaling technique. Since the average delay is inversely proportional to the voltage, the power consumption is proportional to the square of the voltage [see (5), where vdd represents the voltage], so it can be used to estimate the power consumption of these implementations

$$\begin{aligned} delay &\propto vdd^{-1} \\ power &\propto vdd^{2}. \end{aligned} \tag{5}$$

The power consumption experiment results with AVIRIS Indian Pines and ROSIS University of Pavia data sets are presented in the following sections.

1) Results With AVIRIS Indian Pines Data Set: Table IV shows the unit delay, clock cycle, average delay, speedup, power, and power saving percentage of approximate accumulators compared with the traditional RCA when applied to AVIRIS Indian Pines data set. The average delay and power consumption of traditional RCA are 1.23 ns and 0.212 mW, respectively. The average delay, speedup, and power consumption of the approximate accumulator with zero removed error signal are 0.79 ns, 1.56, and 0.1142 mW, respectively. After increasing the number of removed error signals, the average delay, speedup, and power consumption results are 0.56 ns, 2.2, and 0.0637 mW, respectively. This means that around 70% power saving percentage can be achieved.

2) Results With ROSIS University of Pavia Data Set: Table V shows the unit delay, clock cycle, average delay, speedup, power, and power saving percentage of approximate accumulators compared with the traditional RCA when applied to ROSIS University of Pavia data set. The average delay and power consumption of traditional RCA are 1.69 ns and 0.309 mW, respectively. The average delay, speedup, and power consumption of approximate accumulator with two removed error signal are 0.74 ns, 2.30, and 0.0771 mW, respectively. The average delay, speedup, and power consumption of approximate accumulator with four removed error signal are 0.64 ns, 2.63, and 0.063 mW, respectively. According to the accuracy analysis in Section IV-B, when the number of removed error signals is less than three, individual class values are almost the same with regards to the precise computation results. That means that around 75% power saving percentage can be achieved with negligible degeneration of classification accuracy. In some

TABLE IV
POWER SAVING PERCENTAGE OF APPROXIMATE ACCUMULATORS COMPARED WITH THE TRADITIONAL RCA WHEN APPLIED TO AVIRIS INDIAN PINES DATA SET

Removed error signals	Unit delay (NS)	Clock cycle	Average delay (NS)	Speedup	Power (MW)	Power saving percentage (%)
0	0.72	1.093	0.79	1.56	0.1142	46
1	0.72	1.077	0.78	1.59	0.1101	48
2	0.69	1.041	0.72	1.71	0.0965	54
3	0.63	1.011	0.64	1.93	0.0759	64
4	0.56	1.000	0.56	2.20	0.0637	70
Traditional RCA	1.23	1.000	1.23	–	0.2120	–

TABLE V
POWER SAVING PERCENTAGE OF APPROXIMATE ACCUMULATORS COMPARED WITH THE TRADITIONAL RCA WHEN APPLIED TO ROSIS UNIVERSITY OF PAVIA DATA SET

Removed error signals	Unit delay (NS)	Clock cycle	Average delay (NS)	Speedup	Power (MW)	Power saving percentage (%)
0	0.70	1.142	0.80	2.11	0.0856	72
1	0.71	1.126	0.80	2.11	0.0943	69
2	0.68	1.083	0.74	2.30	0.0771	75
3	0.66	1.033	0.68	2.48	0.0664	79
4	0.64	1.003	0.64	2.63	0.0630	80
Traditional RCA	1.69	1.000	1.69	–	0.3090	–

application scenarios, if the accuracy decrease is acceptable with four removed error signals, up to 80% power saving percentage can be achieved.

Considering the experimental results reported for the AVIRIS Indian Pines and the ROSIS University of Pavia data sets, the proposed approximate-computing implementations obtain comparable classification accuracies and around 70% power saving percentage compared with the precise computing results based on traditional RCA implementation. It should be noted that the power saving is obtained by comparing approximate accumulators with the corresponding traditional RCA precise implementation. For an entire application, generally, it includes control-intensive resource management code that needs to be executed on error-free hardware and data-intensive computation that may be more error tolerant. Even data-intensive computation procedures may also include precise computing parts and approximate computing parts. Thus, the power savings with regard to a whole application system depend on the proportion of error-tolerant parts employed approximate techniques in the whole application system. In this case study, the SVM classifier system consumes 0.671 mW (measured by $PrimeTime$ tool), and the traditional RCA consumes 0.212 mW when applied to AVIRIS Indian Pines data set. Thus, the proportion of traditional RCA in SVM classifier system is 31.5%. Then the power saving with regard to a whole classifier system can be estimated by $31.5\% \times 70\%$, which is about 22%.

From the analysis reported in this section, it can be seen that with an increase in the number of removed error signals, the deviation between approximate computing and precise computing results will increase slowly; in other words, the corresponding power consumption will decrease and higher power saving percentage can be achieved. The magnitude of output quality degradation and power saving percentage depends on practical application scenarios. For the SVM hyperspectral image classification case study considered in this work, the approximate computing scheme achieves comparable processing accuracy with a considerable proportion of energy saving around 70% when performing the kernel accumulation implementations. As a result, approximate computing of remotely sensed data shows significant advantages and great potential as compared with the traditional processing techniques.

V. CONCLUDING REMARKS AND FUTURE RESEARCH LINES

A novel approximate computing framework for remote-sensing data processing has been presented in this paper. SVM classification of hyperspectral data has been illustrated as a case study of our proposed framework. Experimental results demonstrate that the proposed approximate computing scheme can achieve around 70% power savings in the kernel accumulation computation procedure, as compared to the traditional RCA precise computation, with negligible degradation in classification accuracy. This is an important achievement to meet the restrictions of onboard processing scenarios. Although SVM classification has been used as an illustrative case study, in this work, the proposed framework is general and can be applied to high performance or energy efficient implementations of other remotely sensed data processing algorithms. This is because remote-sensing images often exhibit a large amount of redundancy. In future developments, implementations of other techniques for hyperspectral data processing (such as spectral unmixing) will be developed using approximate computing techniques.

ACKNOWLEDGMENT

The authors would like to thank Prof. D. Landgrebe for making the AVIRIS Indian Pines hyperspectral data set available to the community, Prof. P. Gamba for providing the ROSIS data over Pavia, Italy, along with the training and test sets. Last but not least, the authors would also like to thank the Associate

Editor who handled our paper and the three anonymous reviewers for providing truly outstanding comments and suggestions that significantly helped us to improve the technical quality and presentation of our paper.

REFERENCES

[1] A. F. H. Goetz, G. Vane, J. E. Solomon, and B. N. Rock, "Imaging spectrometry for earth remote sensing," *Science*, vol. 228, pp. 1147–1153, 1985.

[2] S. Jia, Y. Xie, G. Tang, and J. Zhu, "Spatial spectral combined sparse representation based classification for hyperspectral imagery," *Soft Comput.*, vol. 18, pp. 1–10, 2014 [Online]. Available: http://link.springer.com/article/10.1007%2Fs00500-014-1505-4

[3] Y. Qian, F. Yao, and S. Jia, "Band selection for hyperspectral imagery using affinity propagation," *IET Comput. Vis.*, vol. 3, no. 4, pp. 213–222, Dec. 2009.

[4] A. Plaza *et al.*, "Recent advances in techniques for hyperspectral image processing," *Remote Sens. Environ.*, vol. 113, pp. S110–S122, 2009.

[5] A. Plaza, Q. Du, Y. Chang, and R. L. King, "High performance computing for hyperspectral remote sensing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 3, pp. 528–544, Sep. 2011.

[6] C. A. Lee, S. D. Gasster, A. Plaza, C. I. Chang, and B. Huang, "Recent developments in high performance computing for remote sensing: A review," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 3, pp. 508–527, Sep. 2011.

[7] S. Lopez, P. Horstrand, G. M. Callico, J. F. Lopez, and R. Sarmiento, "A novel architecture for hyperspectral endmember extraction by means of the modified vertex component analysis (MVCA) algorithm," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 6, pp. 1837–1848, Dec. 2012.

[8] B. Yang, M. Yang, A. Plaza, L. Gao, and B. Zhang, "Dual mode FPGA implementation of target and anomaly detection algorithms for real-time hyperspectral imaging," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2950–2961, Jun. 2015.

[9] Y. Tarabalka, T. V. Haavardsholm, I. Kasen, and T. Skauli, "Real-time anomaly detection in hyperspectral images using multivariate normal mixture models and GPU processing," *J. Real-Time Image Process.*, vol. 4, pp. 287–300, 2009.

[10] Y. Wu, L. Gao, B. Zhang, H. Zhao, and J. Li, "Real-time implementation of optimized maximum noise fraction transform for feature extraction of hyperspectral images," *J. Appl. Remote Sens.*, vol. 8, p. 084797, 2014.

[11] Y. Wu, J. Li, L. Gao, X. Tan, and B. Zhang, "Graphics processing unit-accelerated computation of the Markov random fields and loopy belief propagation algorithms for hyperspectral image classification," *J. Appl. Remote Sens.*, vol. 9, no. 1, p. 097295, 2015.

[12] M. I. Castillo, J. C. Fernandez, F. D. Igual, and A. Plaza, "Hyperspectral unmixing on multicore DSPs: Trading off performance for energy," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2297–2304, Jun. 2014.

[13] A. Remon, S. Sanchez, S. Bernabe, E. S. Quintana-Orti, and A. Plaza, "Performance versus energy consumption of hyperspectral unmixing algorithms on multi-core platforms," *EURASIP J. Adv. Signal Process.*, vol. 68, pp. 1–15, 2013.

[14] R. Hegde and N. R. Shanbhag, "Energy-efficient signal processing via algorithmic noise-tolerance," in *Proc Int. Symp. Low power Electron. Des.*, 1999, pp. 30–35.

[15] D. Mohapatra, "Approximate computing: Enabling voltage over-scaling in multimedia applications," Ph.D. dissertation, Purdue Univ., West Lafayette, IN, USA, 2011.

[16] L. Leem, H. Cho, J. Bau, Q. A. Jacobson, and S. Mitra, "ERSA: Error resilient system architecture for probabilistic applications", in *Proc. Des. Autom. Test Eur. Conf. Exhib. (DATE'10)*, 2010, pp. 1560–1565.

[17] V. K. Chippa, D. Mohapatra, A. Raghunathan, K. Roy, and S. T. Chakradhar, "Scalable effort hardware design: Exploiting algorithmic resilience for energy efficiency," in *Proc. 47th ACM/IEEE Design Autom. Conf. (DAC'10)*, 2010, pp. 555–560.

[18] D. Mohapatra, V. K. Chippa, A. Raghunathan, and K. Roy, "Design of voltage-scalable meta-functions for approximate computing," in *Proc. Des. Autom. Test Eur. Conf. Exhib. (DATE'11)*, 2011, pp. 1–6.

[19] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[20] J. Xie, K. Hone, W. Xie, X. Gao, Y. Shi, and X. Liu, "Extending twin support vector machine classifier for multi-category classification problems," *Intell. Data Anal.*, vol. 17, no. 4, pp. 649–664, 2013.

[21] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 6, pp. 1351–1362, Jun. 2005.

[22] H. Cho, L. Leem, and S. Mitra, "Ersa: Error resilient system architecture for probabilistic applications," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 4, pp. 546–558, Apr. 2012.

[23] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *Proc. IEEE 18th Eur. Test Symp. (ETS'13)*, 2013, pp. 1–6.

[24] X. Yang, F. Qiao, C. Liu, and H. Yang, "Design of variable latency adder based on present and transitional states prediction," in *Proc. IEEE 23rd Int. Workshop Power Timing Model. Optim. Simul. (PATMOS)*, 2013, pp. 120–125.

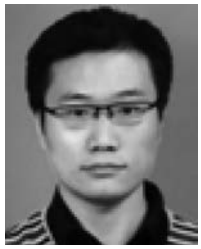
[25] X. Yang, F. Qiao, C. Liu, Q. Wei, and H. Yang, "Design of multi-stage latency adders using detection and sequence-dependence between successive calculations," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS'14)*, 2014, pp. 998–1001.

[26] A. A. D. Barrio, R. Hermida, S. O. Memik, J. M. Mendas, and M. C. Molina, "Multispeculative addition applied to datapath synthesis," *IEEE Trans. CAD Integr. Circuits Syst.*, vol. 31, no. 12, pp. 1817–1830, Dec. 2012.

[27] R. G. Congalton, "A review of assessing the accuracy of classification of remotely sensed data," *Remote Sens. Environ.*, vol. 37, no. 1, pp. 35–46, 1991.

[28] R. G. Congalton, *Kass Green, Assessing the Accuracy of Remotely Sensed Data: Principles and Practices*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2008.

[29] G. M. Foody, "Status of land cover classification accuracy assessment," *Remote Sens. Environ.*, vol. 80, no. 1, pp. 185–201, 2002.



Yuanfeng Wu (M'16) received the B.S. and M.S. degrees in computer science from China University of Mining and Technology, Beijing, China, and the Ph.D. degree in cartography and geographical information system from the Graduate University of Chinese Academy of Sciences, Beijing, China, in 2010.

He is currently an Associate Professor with the Key Laboratory of Digital Earth Science, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, China. His research interests include the development of high-performance computing algorithms and computer software systems in hyperspectral image processing.

Xinghua Yang was born in Datong, Shanxi Province, in 1988. He received the B.S. degree in electronic engineering from Beijing University of Post and Telecommunications, Beijing, China, in 2011. He has been pursuing the Ph.D. degree in circuit and system at the Department of Electronic Engineering, Tsinghua University, Beijing, China, since 2011.

His research interests include variable latency arithmetic unit and approximate computing for high-energy efficiency design.

Antonio Plaza (M'05–SM'07–F'15) was born in Caceres, Spain, in 1975. received the Computer Engineer degree in 1997, the M.S. degree in 1999, and the Ph.D. degree in 2002, all in Computer Engineering from the University of Extremadura, Caceres, Spain.

He is an Associate Professor (with accreditation for Full Professor) with the Department of Technology of Computers and Communications, University of Extremadura, Cáceres, Spain, where he is the Head of the Hyperspectral Computing Laboratory (HyperComp), one of the most productive research groups working on remotely sensed hyperspectral data processing worldwide. He has authored more than 500 publications, including 152 journal papers (more than 100 in IEEE journals), 22 book chapters, and more than 240 peer-reviewed conference proceeding papers (94 in IEEE conferences). He has edited a book on *High-Performance Computing in Remote Sensing* (CRC Press/Taylor & Francis) and

guest edited 9 special issues on hyperspectral remote sensing for different journals. He has been the Advisor of 12 Ph.D. dissertations and more than 30 M.Sc. dissertations. His research interests include remotely sensed hyperspectral imaging, pattern recognition, signal and image processing, and efficient implementation of large-scale scientific problems on parallel and distributed computer architectures.

Dr. Plaza is an Associate Editor for the IEEE ACCESS and was a member of the Editorial Board of the IEEE Geoscience and Remote Sensing Newsletter (2011–2012) and the IEEE GEOSCIENCE AND REMOTE SENSING MAGAZINE (2013). He was also a member of the Steering Committee of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS and REMOTE SENSING (JSTARS). He was the Coordinator of the Hyperspectral Imaging Network, a European project with total funding of 2.8 million Euro. He served as the Director of Education Activities for the IEEE Geoscience and Remote Sensing Society (GRSS) from 2011 to 2012, and has been serving as a President of the Spanish Chapter of IEEE GRSS (since November 2012). He has served as a Proposal Evaluator for the European Commission, the National Science Foundation, the European Space Agency, the Belgium Science Policy, the Israel Science Foundation, and the Spanish Ministry of Science and Innovation. He has reviewed more than 500 manuscripts for over 50 different journals. He is currently serving as the Editor-in-Chief of the IEEE Transactions on Geoscience and Remote Sensing journal. He was a recipient of the recognition of Best Reviewers of the IEEE Geoscience and Remote Sensing Letters (in 2009) and recognition of Best Reviewers of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (in 2010), a journal for which he served as Associate Editor from 2007 to 2012. He was also a recipient of the 2013 Best Paper Award of the JSTARS journal, the most highly cited paper (2005–2010) in the *Journal of Parallel and Distributed Computing*, the Best Paper Awards at the IEEE International Conference on Space Technology and the IEEE Symposium on Signal Processing and Information Technology, the Best Ph.D. Dissertation Award at University of Extremadura, and a recognition also received by six of his Ph.D. students.

Fei Qiao received the B.S. degree in electronic engineering from Lanzhou University, Lanzhou, China, in July 2000, the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2006.

He is currently a Faculty Member of NICS (Nano Scale Integrated Circuits and Systems) Laboratory, Tsinghua University. His research interests include computing architectures for energy-efficient integrated image signal processing and machine learning systems with novel devices, circuit topologies and methodologies.

Lianru Gao (M'12) received the B.S. degree in civil engineering from Tsinghua University, Beijing, China, in 2002, and the Ph.D. degree in cartography and geographic information system from the Institute of Remote Sensing Applications, Chinese Academy of Sciences (CAS), Beijing, China, in 2007.

He is currently a Professor with the Key Laboratory of Digital Earth Science, Institute of Remote Sensing and Digital Earth, CAS. He has authored more than 90 papers in China and abroad. His research interests include hyperspectral image processing and information extraction.

Bing Zhang (M'11–SM'12) received the B.S. degree in geography from Peking University, Beijing, China, in 1991, the M.S. and Ph.D. degrees in remote sensing from the Institute of Remote Sensing Applications, Chinese Academy of Sciences (CAS), Beijing, China, in 1994 and 2003, respectively.

Currently, he is a Full Professor and the Deputy Director of the Institute of Remote Sensing and Digital Earth (RADI), CAS, where he has been leading key scientific projects in the area of hyperspectral remote sensing for more than 20 years. He has been a Professor with the University of Chinese Academy of Sciences, Beijing, China, and gave lessons of Hyperspectral Remote Sensing for more than 10 years. He has developed five software systems in the image processing and applications. He has authored more than 300 publications, including 171 journal papers. He has edited six books/contributed book chapters on *Hyperspectral Image Processing and Subsequent Applications*, those books are serve as the main materials for education and research in hyperspectral remote sensing in China. His research interests include the development of mathematical and physical models and image processing software for the analysis of hyperspectral remote-sensing data in many different areas, such as geology, hydrology, ecology, and botany.

Dr. Zhang has been serving as an Associate Editor for the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (IEEE JSTARS) since 2011, he is also the Guest Editor for the IEEE JSTARS special issue on "Hyperspectral Remote Sensing: Theory, Methods, and Applications" in 2013, IEEE JSTARS special issue on "Big Data in Remote Sensing" in 2015, and *Pattern Recongition Letters* (PRL) special issue on "Advances in Pattern Recognition in Remote Sensing" in 2015. He has been served as a Technical Committee Member of the IEEE Workshop on Hyperspectral Image and Signal Processing (IEEE WHISPERS) since 2011, and as the Chair of China National Committee of International Society for Digital Earth since 2012. He was the advisor of 29 Ph.D. dissertations and more than 12 M.S. dissertations. He is the Student Paper Competition Committee Member in IGARSS 2015 and 2016, and Scientific Committee Member of IGRASS in 2014. He was the recipient of the National Science Foundation Award for Distinguished Young Scholars of China in 2013. He was also awarded as one of the top four scientists among more than 100 professors in RADI in 2015 by Chinese Academy of Sciences. His creative achievements were rewarded eight important prizes from Chinese government, and special government allowances of the Chinese State Council.

Yabo Cui was born in Pingdingshan, Henan Province, in 1990. Since 2013, he has been pursuing the M.S. degree in computer science and technology at the School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China.

He is currently working with the Institute of Remote Sensing and Digital Earth, Chinese Academy of Science, Beijing, China.