A 92-mW Real-Time Traffic Sign Recognition System With Robust Illumination Adaptation and Support Vector Machine

Junyoung Park, *Student Member, IEEE*, Joonsoo Kwon, *Student Member, IEEE*, Jinwook Oh, *Student Member, IEEE*, Seungjin Lee, *Member, IEEE*, Joo-Young Kim, *Member, IEEE*, and Hoi-Jun Yoo, *Fellow, IEEE*

***Abstract*—A low-power real-time traffic sign recognition system that is robust under various illumination conditions is proposed. It is composed of a Retinex preprocessor and an SVM processor. The Retinex preprocessor performs the Multi-Scale Retinex (MSR) algorithm for robust light and dark adaptation under harsh illumination environments. In the Retinex preprocessor, the recursive Gaussian engine (RGE) and reflectance engine (RE) exploit parallelism of the MSR tasks with a two-stage pipeline, and a mixed-mode scale generator (SG) with adaptive neuro-fuzzy inference system (ANFIS) performs parameter optimizations for various scene conditions. The SVM processor performs the SVM algorithm for robust traffic sign classification. The proposed algorithm-optimized small-sized kernel cache and memory controller reduce power consumption and memory redundancy by 78% and 35%, respectively. The proposed system is implemented as two separated ICs in a 0.13-μm CMOS process, and the two chips are connected using network-on-chip off-chip gateway. The system achieves robust sign recognition operation with 90% sign recognition accuracy under harsh illumination conditions while consuming just 92 mW at 1.2 V.**

***Index Terms*—Multiscale Retinex (MSR), network-on-chip (NoC), support vector machine (SVM), traffic sign recognition.**

I. Introduction

OBJECT recognition has been highlighted as a key enabler for various applications demanding visual intelligence such as human friendly interfaces, intelligent robots, security, and automotive vehicle systems [1]–[3]. Among these, the advanced driver assistance systems (ADAS) is receiving much attention since it can increase the level of safety as well as give better experiences to the drivers in vehicle. Recently, many types of technologies have been investigated to support the functions for ADAS. Radar, magnetic referencing, high-accuracy digital maps with GPS, and vision-based techniques are the key supporting techniques for ADAS, and the intelligent cars equipped with these techniques can support forward collision detection, front/rear park assist, and blind spot detection [4]. Among these, the vision-based driver assistance system is the rapidly dominating technique because the video camera is an attractive device from the cost perspective, and it can operate concurrently with radars, which has long ranging capability and azimuth accuracy, to increase accuracy as well as give additional functionalities such as traffic sign recognition, object detection, and classification. However, the vision-based driver assistance system requires high computing power for image processing and pattern recognition, and it is difficult to realize the reliable and robust performance under harsh illumination conditions.

In this paper, we propose a hardware architecture to achieve robust performance under harsh illumination conditions by resolving computationally expensive computer vision algorithms in a low-power budget. Also, we will show the proposed architecture can satisfy several requirements of the traffic sign recognition system for ADAS.

Previous multicore object recognition processors [1]–[3] were able to recognize the traffic signs by using limited computing power. However, they are vulnerable to harsh illumination conditions in driving environment such as abrupt illumination changes in dark tunnels, specular reflections, smear effect, and backlight effect, as shown in Fig. 1. Bad illumination conditions should be taken into consideration in order to make recognition processor show no performance degradation compared with its operation in indoor environments.

The performance of the previous system is degraded due to the limited dynamic range of their image sensors in the cases of rapid changes of light intensity. Conventionally, the sensory issues were addressed to resolve these problems, for example, reducing sensor noise or stabilization of conversion of electrons from photons [5]. Even though these approaches were effective to reduce the effect of sensor noise, there are still some problems such as the shifting of the intensity offset or limited dynamic range.

In order to realize the robust traffic sign recognition overcoming the above problems, we adopt two different algorithms; Multiscale Retinex (MSR) at the front stage and support vector machine (SVM) at the back stage of the recognition algorithmic pipeline. MSR is one of the image enhancement algorithms, and it shows good performance on dynamic range compression and

Manuscript received February 11, 2012; revised May 05, 2012; accepted June 21, 2012. Date of publication September 06, 2012; date of current version October 26, 2012. This paper was approved by Guest Editors Shen-Iuan Liu and Tsung-Hsien Lin. This work was supported by the Global Frontier R&D Program on "Human-centered Interaction for Coexistence" funded by the National Research Foundation of Korea grant funded by the Korean Government (MEST) under Grant NRF-M1AXA003-2011-0028368.

J. Park, J. Oh, S. Lee, and H.-J. Yoo are with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 305-701, Korea (e-mail: junyoung.park@kaist.ac.kr).

J. Kwon is with the Memory Division, Samsung Electronics, Hwasung 445-701, Korea.

J.-Y. Kim is with Microsoft Research, Redmond, WA 98052 USA.

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/JSSC.2012.2211691

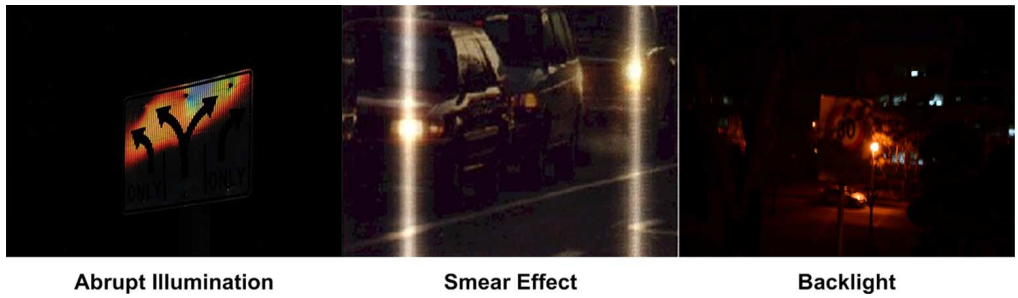


Fig. 1. Harsh illumination conditions in traffic sign recognition.

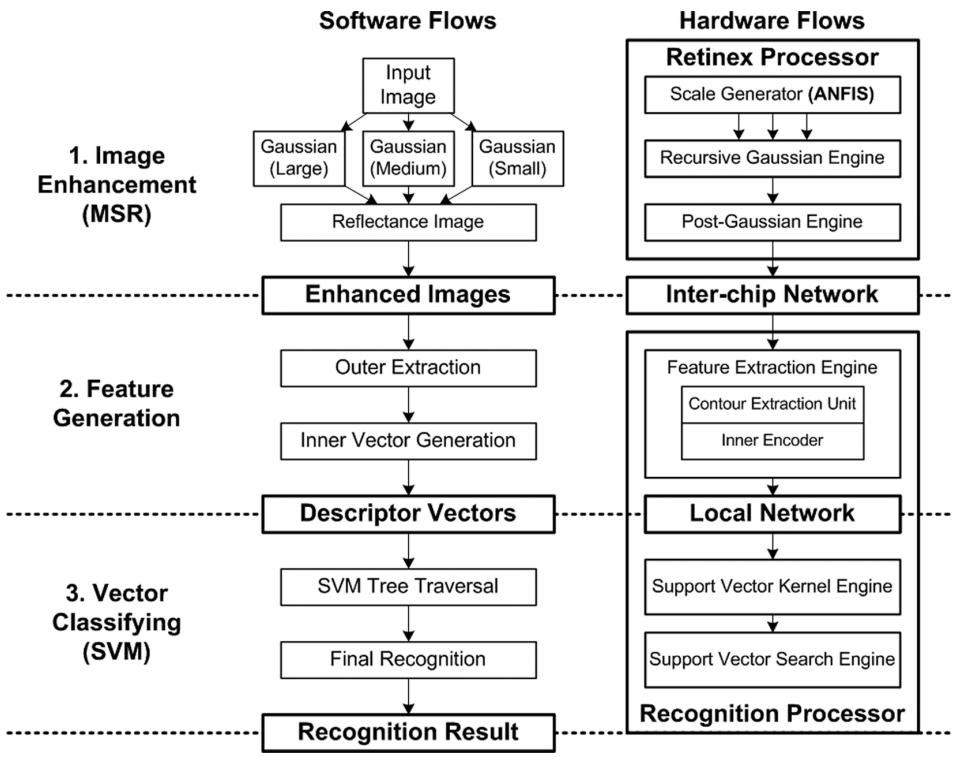


Fig. 2. Overall algorithm and hardware mapping.

color restoration [6]. It is basically based on the idea that human being can easily see individual objects both in the sunlight and in shadowed area, since the eye locally adapts to the abrupt change of light intensity. It improves the contrast, brightness, and perceived sharpness of the input images so that, in the next feature extraction stage of the recognition pipeline, more deterministic features can be obtained for the better object recognition. The SVM is well known for its classification accuracy [7] because it tries to maximize the margin of classification decision rather than to minimize the classification error in other classifiers when training. Even though the algorithms, MSR and SVM, have superior classification accuracy, they were rarely used in the mobile applications due to their demanding requirement of computing power.

In this paper, the proposed system exploits two key features to realize these algorithms for the reliable and robust real-time traffic sign recognition under harsh illumination conditions. First, the Retinex preprocessor is designed to realize the MSR algorithm with recursive Gaussian engine (RGE), reflectance engine (RE), and a mixed-mode scale generator (SG) using adaptive neuro-fuzzy inference system (ANFIS). Second, the SVM processor realizes the SVM algorithm with a support vector kernel engine (SVKE) using an algorithm-optimized kernel cache, and support vector search engine (SVSE) with proposed header combined database structure. The scalable network-on-chip (NOC) interface is proposed to integrate the two chips on a single board by the off-chip gateway to realize 30-frame rate of traffic sign recognition. It provides the communication capacity with 640-MB/s maximum bandwidth and the NOC protocol also supports the connectivity with our previous platforms [1]–[3] so that other platforms can optionally use SVM or MSR for their functional extension.

The remainder of this paper is organized as follows. Section II explains the overall system algorithm. Detailed architectures of the Retinex preprocessor and SVM processor are described in Sections III and IV, respectively. Then, Section V describes the scalable NoC, implementation, and evaluation results of the proposed system. Finally, Section VI summarizes the paper.

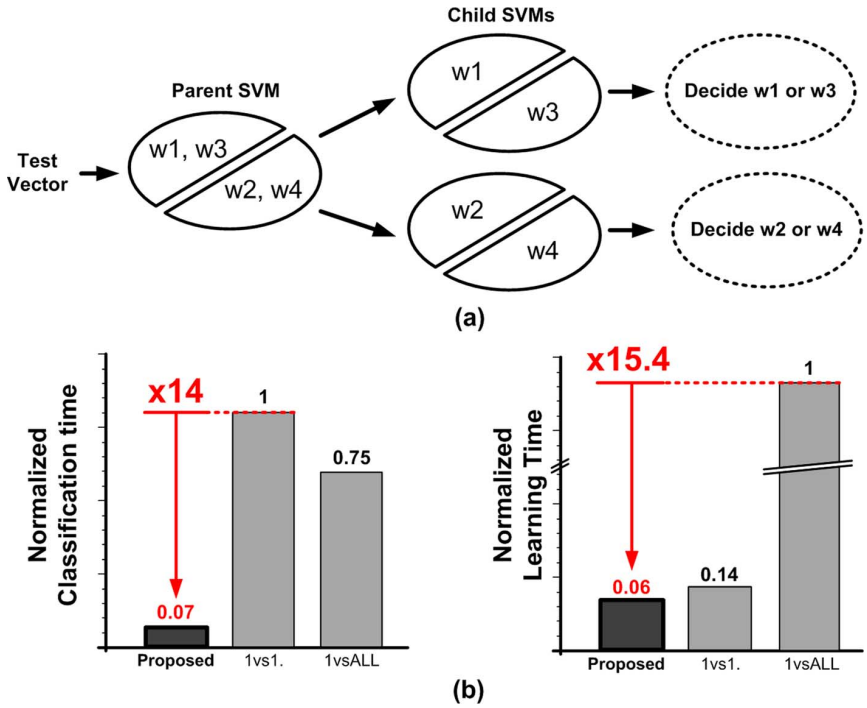


Fig. 3. (a) BDT-based SVM. (b) Performance comparisons of multiclass SVMs.

II. System Architecture

A. Overall Algorithm

In order to achieve reliable and robust traffic sign recognition even for the scenes under dynamically changing illumination conditions, we present an integrated algorithm, which adopts MSR and SVM. The sequence of algorithms explained above and their corresponding hardware are described in Fig. 2. It consists of three functional stages for the robust traffic sign recognition system: image enhancement, feature generation, and vector classification. In the first stage, it performs MSR algorithm which includes three different Gaussian filtering and reflectance image calculation in the Retinex processor. The selection of parameters, Gaussian filtering, and reflectance calculation are performed in separated accelerators, respectively.

Through the inter-chip network connection, the enhanced image from Retinex Processor is now processed in feature generation and vector classification stages. The feature extraction is divided into 2 steps, which are outer extraction and inner vector generation and they are processed in the accelerator which is called Feature Extraction Engine (FFE). In the vector classification stage, Support Vector Kernel Engine (SVKE) and Support Vector Search Engine (SVSE) in the Recognition processor accelerate SVM tree traversal to achieve the real-time SVM operation. After the SVM operation, it finally achieves the recognition result.

Different from previous recognition processors [1]–[3], the proposed system additionally includes the image enhancement stage, the MSR, for the first stage, to adapt harsh illumination changes. This stage generates the enhanced image as a result which helps accurate feature extraction in the next stage. The main idea of the MSR algorithm is that the reflectance or true color of object can be obtained by subtracting the illumination from the perceived brightness, or lightness. The illumination can be deduced from the average value of surrounding pixels. As a result, the basic form of the Retinex algorithm is given by

$$\mathrm{R}_i = \log \mathrm{I}_i(x,y) - \log[F_n(x,y) * I_i(x,y)], \mathrm{i} = 1, \ldots, \mathrm{N} \tag{1}$$

$$F\,(\mathrm{x},\mathrm{y}) = \exp\left[-\frac{x^2+y^2}{\sigma^2}\right] \tag{2}$$

where the subindex i represents each channel of image and N is the total number of channels, for example, $\mathrm{N} = 1$ for a grayscale image and $\mathrm{N} = 3$ for typical color image. The F represents the surround function usually given as the Gaussian function like (2), which is used for calculating the illumination. I and R represent the input and output image, or called reflectance, respectively.

In this algorithm, it is critical to select the appropriate scale parameter σ, which determines the extend amount of the surround for each scene. As σ gets smaller, the detail expression of the image gets sharper, and, on the other hand, bigger σ results in natural color rendition. The problem is how to find the appropriate scale parameter σ for the given scene:

$$\mathrm{R}_{\mathrm{MSRi}} = \sum_{n=1}^{3} w_n \{\log I_i\,(x,y) - \log[F_n\,(x,y) * I_i\,(x,y)]\}. \tag{3}$$

MSR, as shown in (3), uses three different kinds of the scale parameters and sums up the corresponding images to achieve a balance between dynamic range compression and tonal rendition whereas Retinex uses only one scale parameter. However, the problem still remains because the values of three scale parameters are still too sensitive to the scene variation. In this

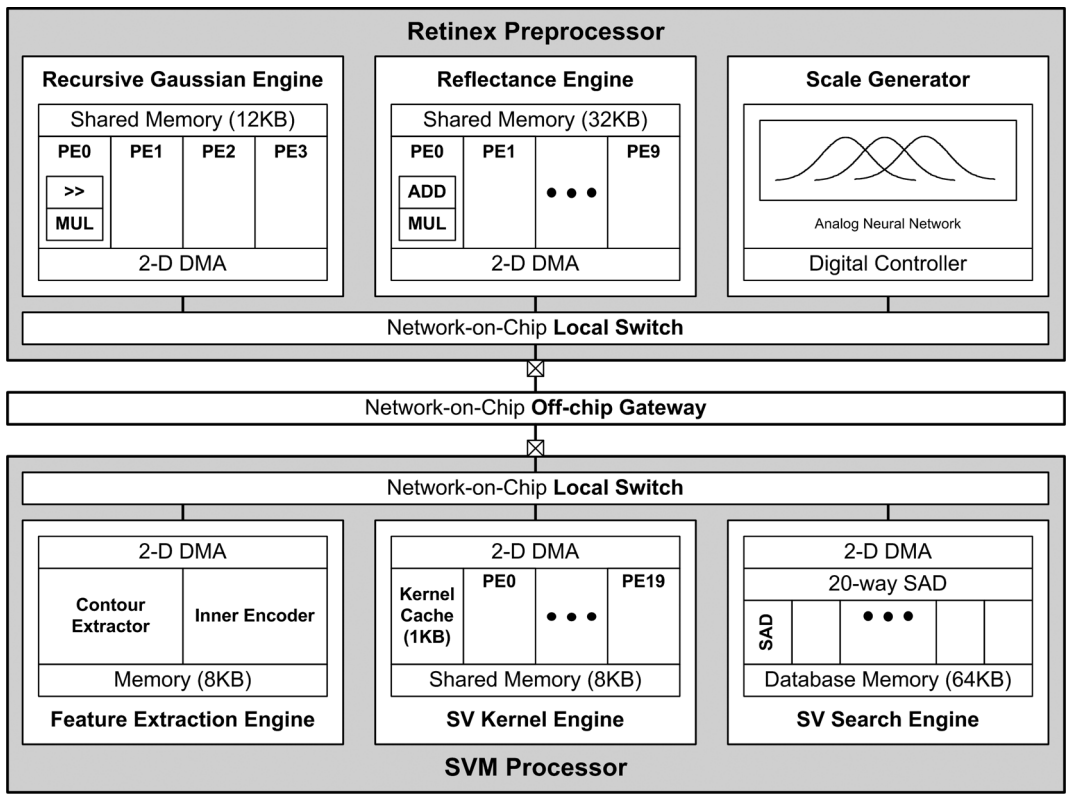


Fig. 4. Block diagram of the proposed traffic sign recognition system.

work, we use neuro-fuzzy inference algorithm with three types of statistical data from image to estimate the optimal scale parameters without any manual tuning. It is performed by the mixed-mode Adaptive Neuro-Fuzzy Inference System (ANFIS) and will be explained later in Section III.

After then, the feature extraction is performed in two steps based on the MSR processed image: outer extraction and inner vector generation. Because the traffic signs are designed for easy identification using different symbols, colors and shapes, the contour extraction stage generates the region-of-interest (ROI) for feature description based on these specific characteristics by template matching. Then, the inner vector description using scale-invariant feature transform (SIFT) [8] extracts some features from a local image region. In general, the traffic sign comprises only a part of the entire image and the SIFT description is performed in this small region. Therefore, in this work, it is possible to perform the SIFT-based feature description in real-time without time overhead.

Finally, the extracted vectors are classified by SVM. SVM takes a set of input vectors, which are called as support vectors (SVs), so that the vectors of the separate classes can be divided by a maximum margin that is as wide as possible. Since SVM is a binary classifier, multiple SVMs should be used together in order to classify N multicategories. A variety of techniques for classifying a multicategory classification problem using binary classifiers have been proposed [9]–[11]. There are three representative methods of forming multi-category classifier using SVM: 1-vs-ALL, 1-vs-1, and binary decision tree (BDT). In 1-vs-ALL method, it constructs N two-class SVM classifiers for a N-class problem. Each SVM is trained while labeling the samples in one class as positive and the remaining samples as negative. In the 1-vs-1 method, it constructs $\mathrm{N}(\mathrm{N}-1)/2$ two-class SVM classifiers for a N-class problem. It use all the binary pair-wise combinations of the N classes, and each SVM is trained while labeling the samples in one class as positive and the sample in another class as negative. In the BDT method, it constructs a binary decision tree while each node of the tree is SVM which divides the samples into two categories. Among several multi-class implementations, the BDT-SVM has relatively low complexity as $\mathrm{O}(\log_2 \mathrm{N})$ compared with 1-vs-1 and 1-vs-all, which have the complexity of $\mathrm{O}(\mathrm{N}^2)$ and $\mathrm{O}(\mathrm{N})$, respectively [11].

In this work, the BDT-based SVM is adopted for multi traffic sign recognition. Fig. 3(a) shows the algorithm flow of the adopted BDT-based multiclass SVM. By adopting the BDT-based SVM and optimized learning techniques, the classification and learning time are 14 times faster and 15.4 times faster, respectively, than conventional learning algorithms as shown in Fig. 3(b).

B. Overall Architecture

Fig. 4 shows the overall block diagram of the proposed system, which performs the proposed traffic sign recognition algorithm combining MSR and SVM. The Retinex preprocessor performs MSR algorithm for illumination adaptation, and SVM processor performs SIFT-based feature extraction and SVM classification. The Retinex consists of the recursive Gaussian engine (RGE), Reflectance Engine (RE), and a mixed-mode

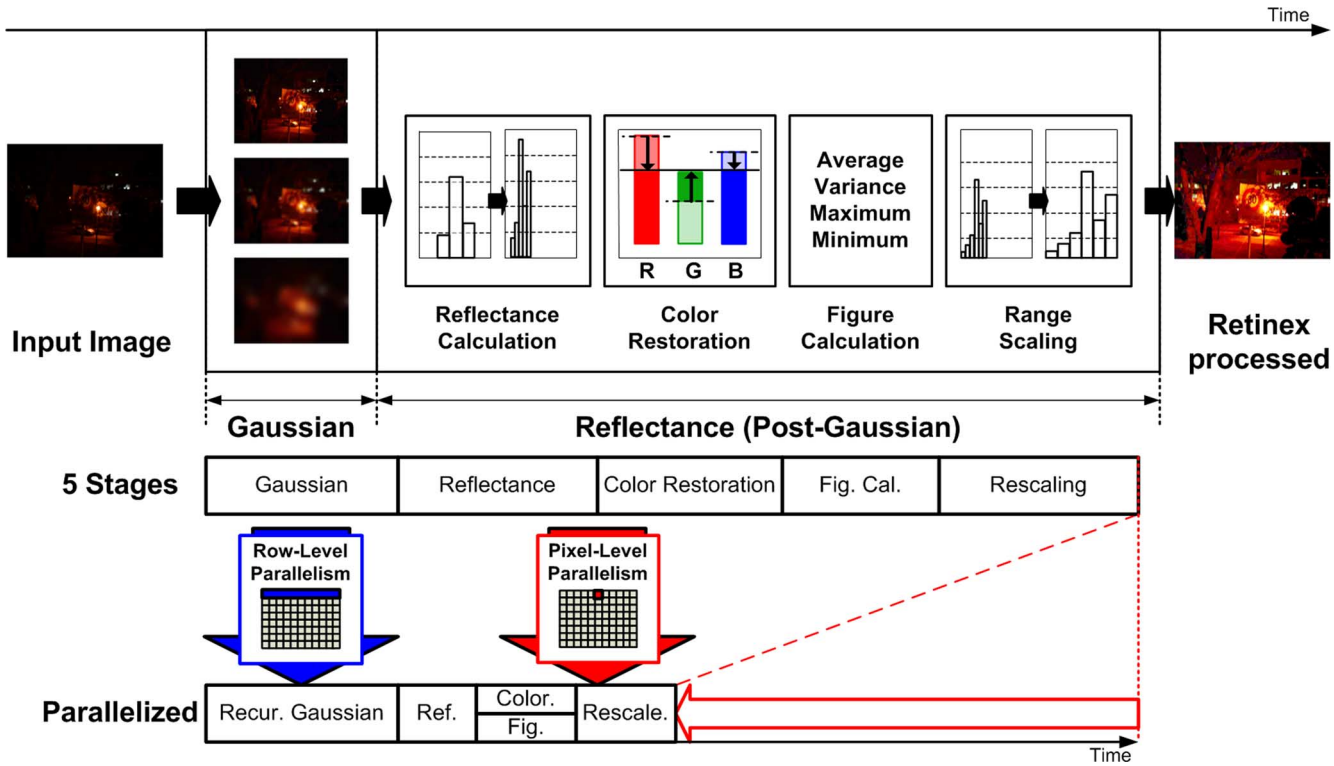


Fig. 5. Two-stage pipelined architecture for MSR implementation.

scale generator (SG) with ANFIS [12]. SG measures the statistical data of the given image, which is down-sampled to 80×60 and classifies it into one of the illumination levels by fuzzy inference. Then, SG determines the scale parameter according to the classified illumination level. The content-aware neuro-fuzzy inference in SG brings the better feature extraction result by deciding the scale parameter of MSR algorithm automatically.

The MSR algorithm is organized into two stages, Gaussian filtering and Reflectance acquisition, which will be explained in detail in Section III. RGE performs recursive Gaussian filtering with low hardware overhead as the parameters of the Gaussian filter are changing [13]. After Gaussian filtering, RE executes the reflectance acquisition. The execution times of RGE and RE are adjusted equally by performing Gaussian filtering in the recursive method and executing Reflectance acquisition in paralleled.

The generated image after performing MSR algorithm is transferred to the SVM processor for the feature extraction and recognition of the traffic signs. The SVM processor consists of the feature extraction engine (FEE), the SVKE, and the SVSE as shown in the bottom of Fig. 4. The FEE extracts only the traffic signs from the image and generating SIFT-based feature vectors. FEE again consists of two blocks: contour extractor (CE) and inner encoder (IE). The CE selects the region of the traffic sign while rejecting regions containing the other distractors. In the CE, the outer edge of a specific color, such as red or blue, is traced and only ROIs of the traffic sign are segmented in a tile-based approach [3]. The IE generates the SIFT-based feature descriptors of the segmented ROIs. After

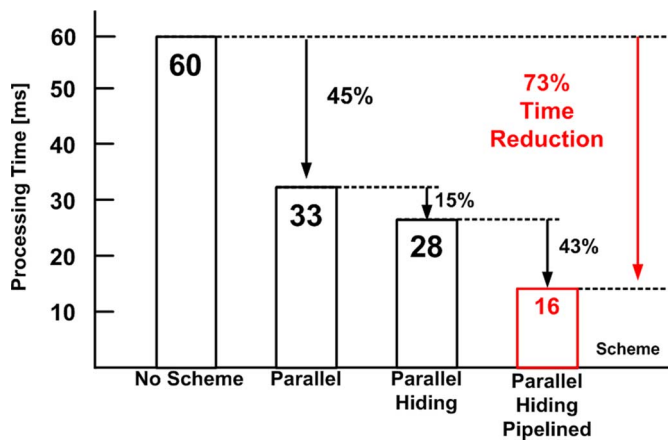


Fig. 6. Processing time reduction by two-stage pipelined architecture.

the feature description is completed, the feature vector is classified by SVKE, which performs SVM algorithm for feature vector classification. In order to implement SVM function to meet the real-time requirement, a lookup-table (LUT) kernel cache is adopted in SVKE. SVSE controls the massive call of the support vectors from the SVKE with the 64-bit header and 160-bit attributes, and logics.

The Retinex preprocessor and SVM processor are implemented in separate ICs and communicated through NoC off-chip gateway on the board. The two proposed chips use the same network interface on the off-chip gateway, which has been used in the previous object recognition chips [1]–[3]. The 16-depth first-in–first-out (FIFO)-based synchronization

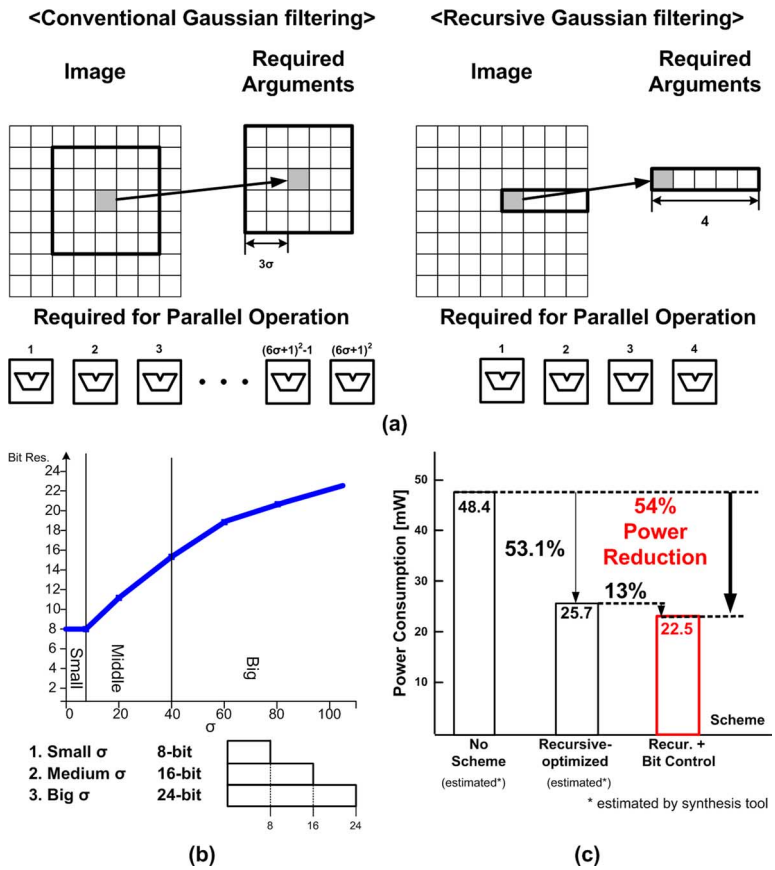


Fig. 7. (a) Gaussian Engine in conventional and recursive method. (b) Required bit resolution. (c) Effectiveness of RGE features.

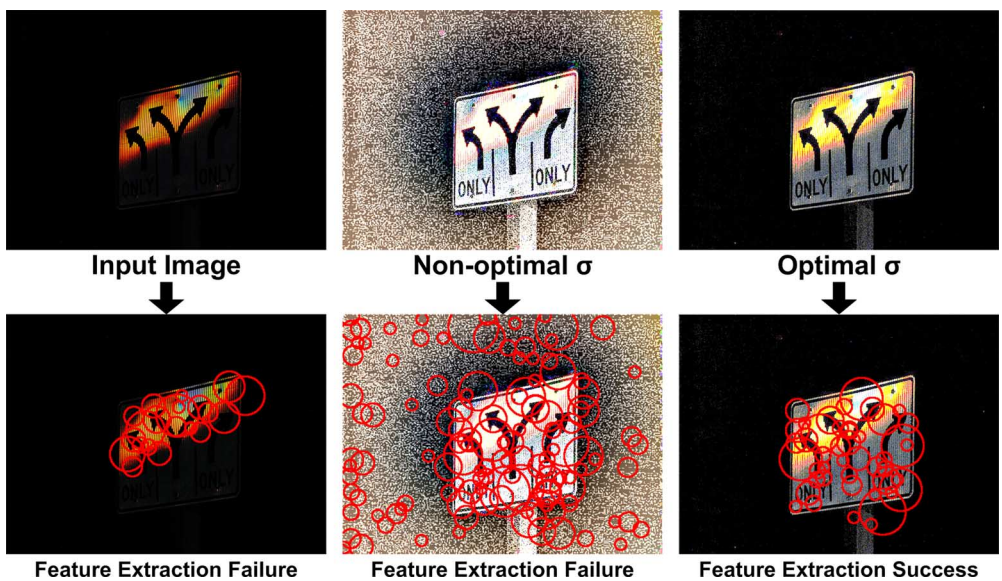


Fig. 8. Effects of the scale parameter on SIFT feature extraction.

switch is redesigned in the top switch for communication stability and bandwidth requirement. The protocol interface of NoC is compatible to [1]–[3] so that the proposed processor can fully communicate with previous object recognition chips for application extension. The details of two processors, Retinex preprocessor and SVM processor, will be explained in Section III.

III. Retinex Preprocessor

A. Two-Stage Pipelined MSR Algorithm

In this study, MSR is divided into two operation stages, Gaussian operation and reflectance. The former is performed by RGE and the latter is performed by RE in the Retinex preprocessor. Gaussian operation can estimate the illumination

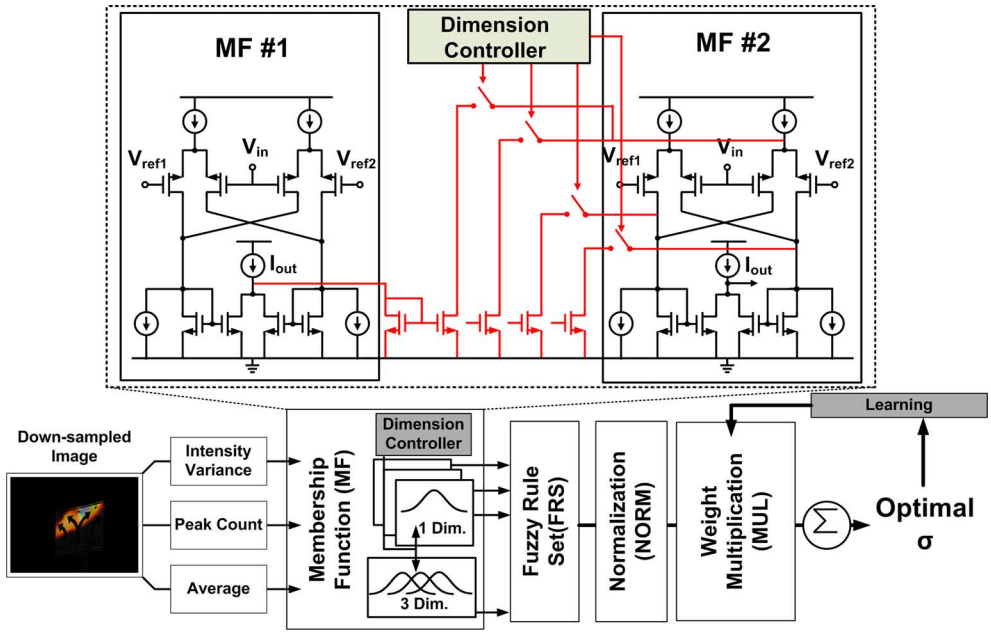


Fig. 9. SG with ANFIS.

of the given image because common feature of the image like illumination can be extracted by Gaussian filtering. Gaussian filtering is performed in the recursive method [13] to achieve the pipelined operation. It is because in the recursive Gaussian model, the filtering operation at a pixel requires only 4 adjacent pixels in a row, that is, there is no data dependency between rows, so that it can be parallelized in row level.

The Reflectance stage, as shown in Fig. 4, also has no data dependency between pixels and can be parallelized in pixel level. The Reflectance stage can be further divided into four substages as shown in Fig. 5. The four substages are composed of Logarithmic calculation, Restoration, Figure calculation, and Range rescaling. First, reflectance of the image is calculated by subtracting the illumination information from the given image. Second, weights of three color channels (R, G, B) are decided. Third, figures of the image such as average, variance, maximum, and minimum are calculated. Finally, the information is used to rescale the resulting image for appropriate image expression.

The number of processing elements in RGE and RE is decided to balance their execution times to achieve the two-level parallelism between Gaussian filtering and Reflectance. The RGE has four processing elements for exploiting row level parallelism of the recursive Gaussian method, and the RE has ten processing elements to use pixel level parallelism of the Reflectance stage in MSR. The execution time of the Reflectance stage is further reduced by performing the color restoration and figure calculation in parallel. Combining all of the features, the execution time of RGE and RE can be overlapped and pipelined by reducing its processing time by 73% in total. Fig. 6 shows the processing time reduction according to the pipelining of stages.

B. Recursive Gaussian Engine

Fig. 7(a) compares the operations between the conventional method and the recursive method [13]. In a conventional

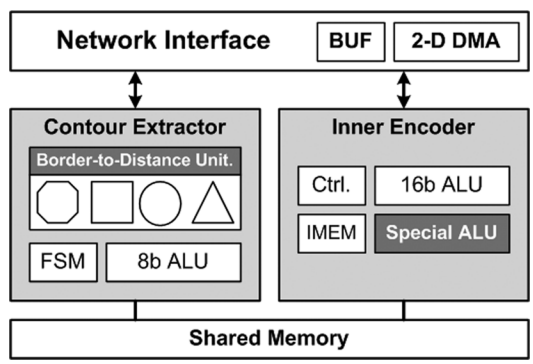


Fig. 10. Feature extraction engine.

method, the radius of the Gaussian filter window is usually determined by three times the scale parameter of the filter in most cases. Considering the fact that MSR algorithm requires several different types of scale values, different sizes of windows are required, or for example, more than 300 kinds of scale parameters are required in QVGA-sized image. The undetermined window size should be avoided in the pipelined operation because the execution time is unexpected; furthermore, it requires $(6\sigma + 1)^2$ multiplications and additions for one pixel calculation so that it is hard to manage the operation in a parallel way. On the other hand, the recursive Gaussian method decomposes Gaussian operation into causal and anticausal filters, which consists of four multiplications along the adjacent pixel values. It dramatically reduces the required memory bandwidth for one pixel calculation because only four concurrent multiplications and additions are required. However, it has an error because it approximates the coefficient values, not the exact values of the true Gaussian filter. In MSR, it is not a serious problem because the algorithm requires only three relatively different Gaussian filters regardless of whether each filter has an accurate Gaussian filter kernel. Even so, we use an adaptive bit-resolution control

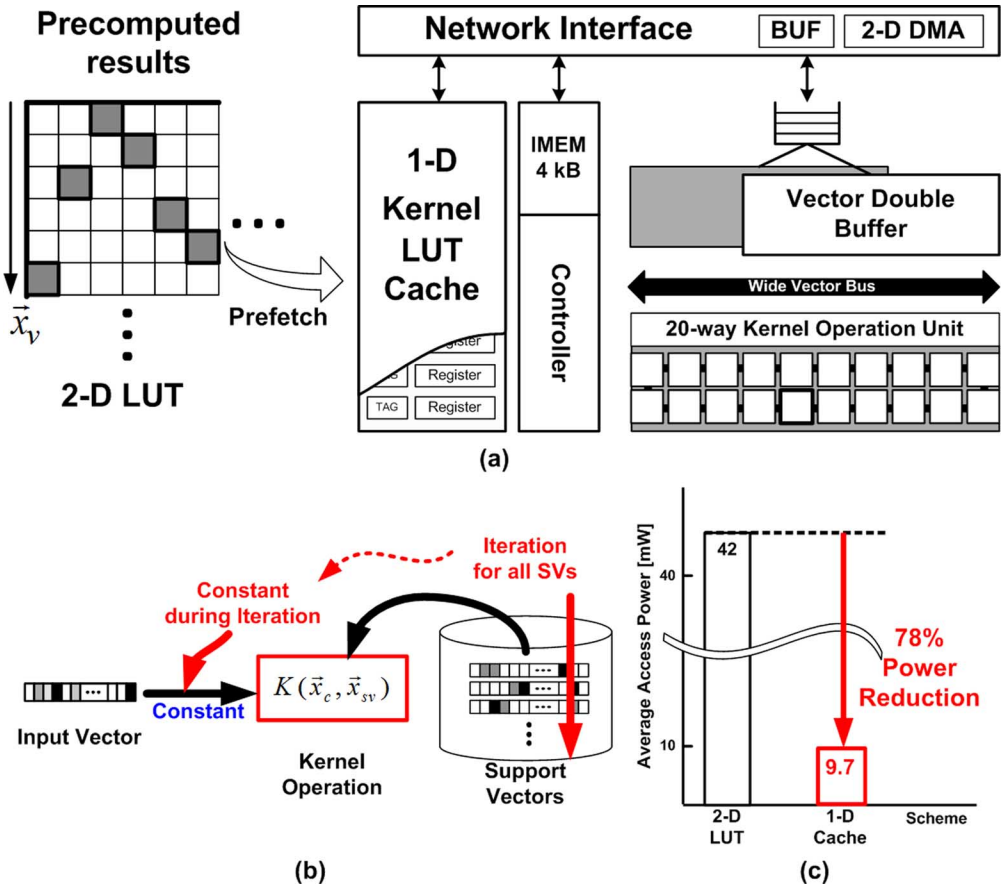


Fig. 11. (a) SVKE. (b) Two parameters of the kernel operation in SVM. (c) Effectiveness of SVKE features.

for better filter operation. Fig. 7(b) shows the required bit resolution for maintaining the precision of the recursive Gaussian method, and it supports three different types of fixed point operation. In this work, 16-bit or 24-bit resolution operation is selectively used according to the scale parameters of the Gaussian filter. With the adaptive resolution, we can reduce the power consumption by turning off the wide-bit processing elements if it is unnecessary. As shown in Fig. 7(c), total power reduction amounts to 54% compared with the conventional implementation results estimated by the synthesis tools.

C. Mixed-Mode SG Using ANFIS

Fig. 8 shows that if the Retinex preprocessor chooses the wrong scale parameter for the Gaussian filter, it fails the feature extraction because of the resulted undesired clutters of MSR algorithm. Since the clutters are closely dependent on the pixel intensity variance of the image, the SG can be used to predict the optimal scale parameter of the Gaussian filter using simple statistics such as intensity variance, peak count, and average. Fig. 9 shows the operation stages in SG which has neuro-fuzzy inference steps and hardware mapping. The SG consists of a mixed-mode ANFIS [12] and a digital controller. By the mixed-mode design, ANFIS takes advantages of low-power operation of the current mode analog circuits resulting in reduce power reduction by 15%.

The mixed-mode ANFIS consists of five stages for neuro-fuzzy inference: membership function calculation, fuzzy rule set calculation, normalization, weight multiplication, and sum up. Among these steps, the analog datapath of the membership function calculation is shown in Fig. 9, which performs nonlinear conversion of the input to fuzzy values. The boundaries of fuzzy values are controlled by V_{ref1} and V_{ref2}. The mixed-mode design of ANFIS was first presented in [14] and has been used for its neuro-fuzzy inferences in several object recognition processors [3], [15]. In RGE, the circuit is revised to support multidimensional operation with dimension controller. A detailed explanation of the circuit operation can be found in [14]. The digital controllers in the other parts perform parameter loading and NoC communication. The variance of the pixel intensities and the peak counts of a down-sampled image at each channel are loaded in the controller, and they provide the clues on the optimal scale parameter for the given scene. In most cases, it is sufficient to predict only the scale parameter since the neuro-fuzzy logic finds the optimal scale values while performing learning operation automatically.

IV. SVM Processor

A. Feature Extraction Engine (FEE)

Fig. 10 shows the detailed block diagram of the CE and IE, which have the finite-state-machine logic and partially programmable controller for their specific jobs – traffic sign detection and description – and several special processing elements, which are border-to-distance calculation unit for traffic

sign segmentation in CE and special ALU for SIFT-feature description in IE. Since the traffic sign has only specific colors, such as red and blue, and specific shapes in general, the CE uses color information and distances from the border to a specific color outer rim of the traffic signs. The integrated 8-bit ALU in CE is capable of performing the segmentation in each color channel with the CE.

After then, the segmented image results are stored in the shared memory to provide them for IE. The IE performs the feature description on the segmented part of the image. It has a partially programmable controller with 4-kBytes instruction memory, 16-bit extended ALU, and special ALU. In this work, scale-invariant feature transform (SIFT) is adopted because it is robust to noise as well as invariant to scale and rotation [8]. The controller supports some special instructions, such as parallel convolution, for SIFT descriptor generation. The CE adopts 8-bit ALU because it handles 8-bit resolution image, whereas the IE adopts 16-bit ALU to meet the algorithm precision requirement of SIFT description. Special functions for SIFT description, such as SQRT, DIV, SIN/COS, and ARCTAN are supported by the special ALU within three cycles.

B. Support Vector Kernel Engine

The generated feature vectors are then classified by SVM classifier, which requires high computational cost due to its complex kernel operation. Fig. 11(a) shows the detailed architecture of SVKE, which performs classification and learning operations of SVM. In this study, we adopt the algorithm-optimized one-dimensional kernel cache in SVKE to reduce the memory size requirement. It can store the pre-computed kernel operation results to enable the complex kernel operation with small memory at one cycle while performing SVM's classification and learning functions. The key functions for SVM classification and learning are shown, respectively, in

$$\mathrm{d}\,(\vec{x}) = \sum_{i \in SV} \overrightarrow{\alpha_i}\,\overrightarrow{y_i}\,k(\overrightarrow{x_i}, \vec{x}) \tag{4}$$

$$\frac{\partial \mathrm{W}(\vec{\alpha})}{\partial \alpha_i} = 1 - \frac{1}{2}\sum_{j=1}^{l} \alpha_j y_i y_j k(\overrightarrow{x_i}, \overrightarrow{x_j}) \tag{5}$$

where α_i is a coefficient, y_i is a class value, x_i is a feature vector, and K denotes kernel operation. In both cases, these functions should be repetitively performed as many times as the number of support vectors, which may be more than 1000 iterations in most cases. Usually, real-time performance is difficult because most of kernel operation have nonlinear functions requiring special floating-point hardware. Most of SVM kernel include Gaussian radial basis function, or hyperbolic tangent functions [7]. An LUT can resolve the problem because it simply loads the precomputed results. However, it requires relatively large memory for mobile system. For the kernel operation with two 8-bit resolution parameters, 64-kBytes memory is required for LUT. For this reason, many studies have implemented the kernels in a limited linear function, 4-bit lower resolution, or logarithmic number system [19]–[23]. However, as shown in (4) and (5), one of the kernel parameters is constant during the iteration. Fig. 11(b) shows the operations of the proposed scheme.

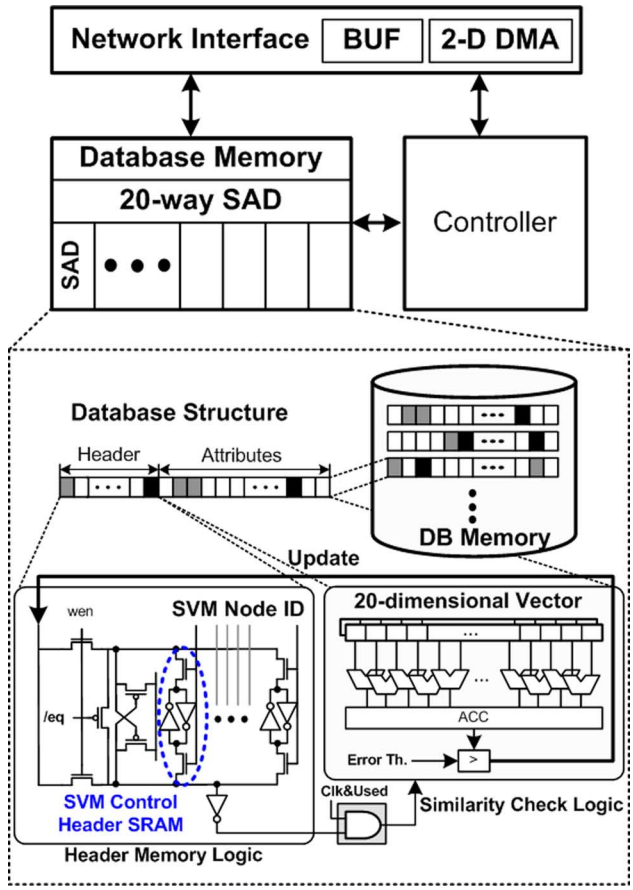


Fig. 12. SVSE.

The input parameters of kernel operation are the input vector and support vectors. When performing the classification, the input vector is constant while the kernel operation is repeated for processing all of the support vectors. Thus, the one-dimensional (1-D) kernel LUT cache fetches the kernel LUT results corresponding to one parameter before performing the kernel operation. It reduces the memory requirements without performance degradation, comparing to the conventional case using the two-dimensional (2-D) LUT. Fig. 11(c) shows the effectiveness of the proposed 1-D kernel cache in comparison to the 2-D kernel cache. Thanks to the reduced cache memory size, the average accessing power consumption is reduced by 78% while maintaining the same throughput.

C. Support Vector Search Engine (SVSE)

Since SVM performs classification based on its support vectors and BDT based multiclass SVM has many common support vectors between parent node and child nodes [9]–[11], it is critical that the common vectors are shared to reduce its requirement of large memory size. Fig. 12 shows the detailed architecture of SVSE and its support vector data format in accordance with its 160-bit datapath of processing elements. Each support vector database consists of the header and attributes. The header part of the database format indicates which SVM has these vector attributes as its own support vector. A binary expression is used, zero for no existence of support vector and

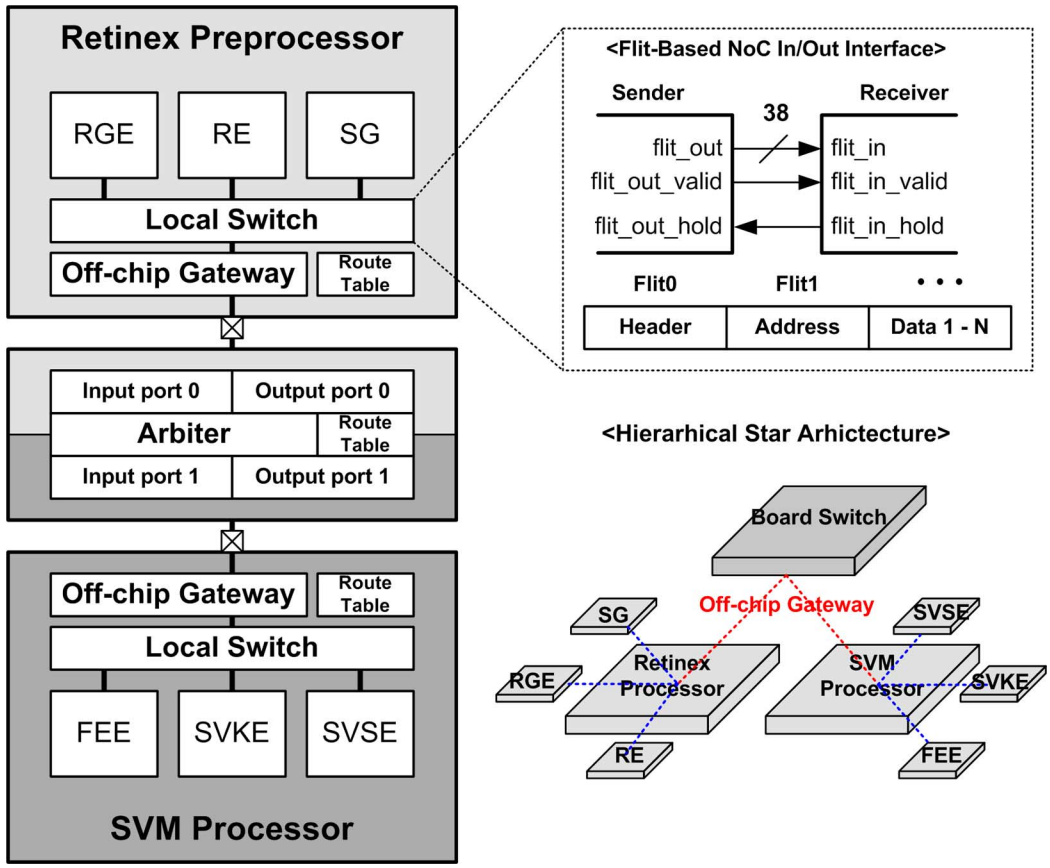


Fig. 13. Top integration using NoC interface.

one for existence of support vector. The header part can indicate up to 64 SVMs. When the controller requests the access of support vector of a SVM, the header memory logic returns the corresponding vectors by Boolean operation. In SVM's learning phase, the similarity check logic reduces the redundancy in memory of duplicated vectors by comparing the differences of vector attributes. When the controller requests to store the given vector as a support vector, the similarity check logic searches the most similar vector in the database. If the difference of the two vectors exceeds the threshold value, the given vector is stored in the database as the new entry. Otherwise, the vector storage is replaced by updating of only the header part of the most similar vector. In the similarity check logic, the input vector is connected directly to the sum of absolute differences (SAD) unit by a 160-bit datapath so that the vector comparison can be performed without overhead.

As a result, it enables about 64 kBytes of internal memory to store more support vectors. In our applications for recognizing 20 traffic signs, 54% of support vectors are duplicated. With this scheme, the total memory requirement can be reduced by 35%.

V. System Implementation

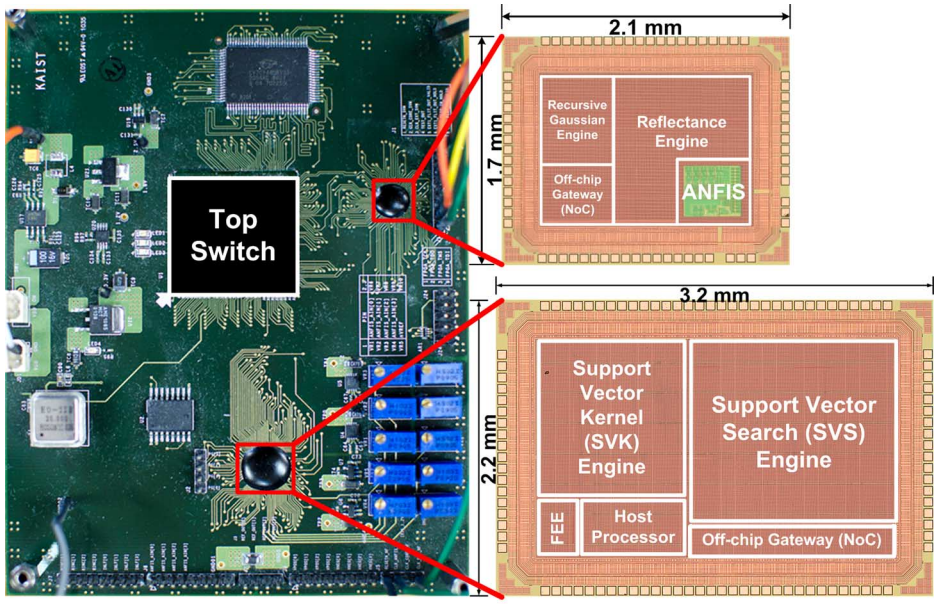
A. NoC Interface for Two-Chip Integration

In this study, a NoC is adopted as a communication method rather than bus, because it provides sufficient bandwidth to each channel and has good scalability by the router switch. For inter-chip communication between Retinex preprocessor and SVM processor, the FIFO-based synchronization switch with 16-depth buffer is included in top switch for communication stability. The NoC interface contains the configurable source routing tables so that the entire routing path can be included in the 38-bit header flit. The packet consists of three types: header, address, and data flits. It supports up to 15 burst operation and, finally, provides 640 MB/s of bandwidth in each direction. Considering that the required bandwidth for Retinex and SVM processors is about 55 MB/s at a QVGA-sized image, it fully supports the full operation and is capable of processing up to VGA-sized image. In this work, we develop the system using a QVGA-sized video interface.

Fig. 13 shows the block diagram of NoC integration of two chips. It uses a wormhole routing protocol that controls the packet by smaller flow unit (flits) to reduce the input buffer size. The two chips are connected on the board in a hierarchical star topology through the off-chip gateway path, as shown in Fig. 13. Because the previous object recognition processors [1]–[3] also have the same off-chip gateway path, it can communicate without any modification.

B. Implementation Results

The proposed traffic sign recognition system is fabricated in a 0.13-μm CMOS technology. The Retinex preprocessor and the SVM processor are implemented in separate ICs and integrated together on an evaluation board through the same NoC protocol. All blocks are implemented by a standard-cell automatic Place and Routing stage except the SG, which is custom-designed



	Retinex Preprocessor	SVM Processor
Technology	0.13 μm 1P8M CMOS	
Area	2.1 mm x 1.7 mm	2.2 mm x 3.2 mm
Frequency	200 MHz IP / 400 MHz NoC	
Power Supply	1.2 V core, 2.5V I/O	
Power Consumption	42 mW	50 mW
Processing Speed	30 fps @ QVGA	

Fig. 14. System photograph and summary.

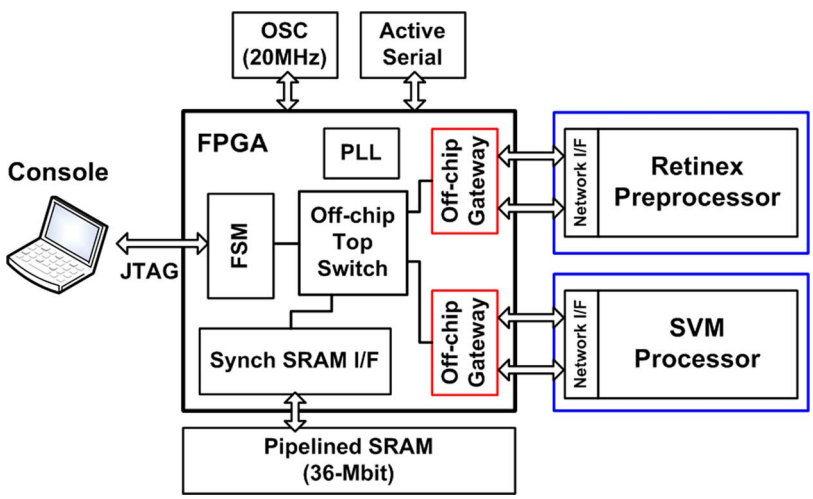


Fig. 15. Block diagram of the evaluation system.

with mixed-mode ANFIS. Fig. 14 shows the chip photograph and summarizes its specifications. The Retinex preprocessor occupies 2.1×1.7 mm^2 consuming 42 mW at a 1.2-V power supply and the SVM processor occupies 2.2×3.2 mm^2 consuming 50 mW at a 1.2-V power supply, respectively. The operating frequency of the processor is 200 MHz for IP blocks and 400 MHz for the NoC interface. The proposed system is capable of performing the traffic sign recognition at QVGA-sized image in 30 fps real time.

C. System Evaluation

The proposed chips are evaluated in a test platform, shown in Fig. 15, using QVGA-sized video that was recorded in a harsh environment at night with irregular illumination of headlight from the front car. Video environment includes abrupt illumination change, under-exposure, and backlighting. The Retinex preprocessor and SVM processor are connected by the off-chip gateway interface in FPGA on the evaluation board. Full traffic

TABLE I
TRAFFIC SIGN RECOGNITION APPLICATION COMPARISONS

	[16]	[17]	[18]	This Work
Illumination Adaptation	No	No	No	Yes
Video	640x480	320x170	656x492	320x240
Accuracy	76 %	98%*	-	90 %
Processing Time	130 ms	85 ms	433 ms	30 ms
Platform	P4 2.7 GHz	PC	FPGA	0.13μm CMOS

* No Dynamic Illumination Condition

TABLE II
SVM FUNCTION COMPARISONS

	[19]	[20]	[21]	[22]	This Work
Kernel Function	Linear	Quadratic	Gaussian	Gaussian	All
On-chip Learning	No	No	Yes	Yes	Yes
Multi-class	No	Yes	No	No	Yes

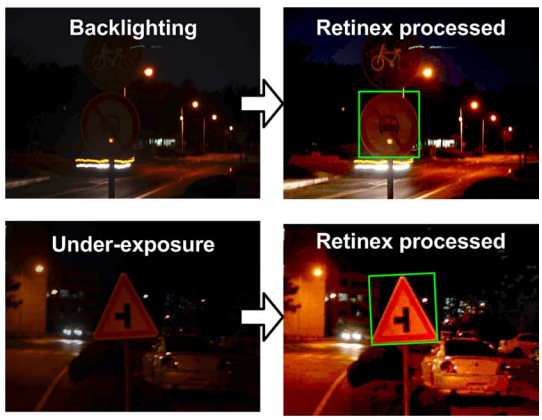


Fig. 16. Traffic sign recognition demonstration.

sign recognition is completed within 30 msec in total while the previous work [16]–[18] required 130 ms, 85 ms or 433 ms on PCs, and FPGAs, respectively. Table I shows the comparisons of the proposed system in traffic sign recognition application. The proposed system can recognize traffic signs reliably and robustly with the help of its illumination adaptation ability while the others lack illumination adaptation. Table II shows the comparison of SVM function capability with other works of SVM processors [19]–[22]. The proposed SVM processor supports all the functions of kernel operation, on-chip learning, and multi-class classification. The recognition accuracy measured in the test video sequences is approximately 90%, which is higher than the 38% recognition accuracy of the previous works [1]–[3] in the harsh illumination conditions of the test video, such as backlighting and under-exposure cases. Thanks to the image enhancement of the Retinex preprocessor and high accuracy of SVM processor, it successfully operates even with limited dynamic range mobile camera sensor as shown in Fig. 16.

VI. CONCLUSION

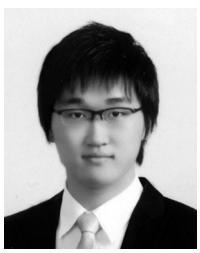
In this paper, we propose a robust traffic sign recognition system for ADAS which can clearly recognize traffic signs even under harsh environments. It is composed of two separate chips, the Retinex preprocessor and the SVM processor. The proposed system supports illumination adaptation capability under abrupt illumination change by implementing MSR algorithm in the Retinex preprocessor and guarantees high classification accuracy by the SVM algorithm in the SVM processor. The Retinex preprocessor consists of RGE, RE, and mixed-mode SG. The MSR operation is parallelized and pipelined into two stages, resulting in 73% processing time reduction while the adopted recursive method achieves 54% power reduction compared with the conventional implementation. The mixed-mode SG decides the parameters automatically by the learning operation of ANFIS. The SVM processor consists of FEE, SVKE, and SVSE blocks. The 1-D kernel cache in SVKE performs the SVM functions with small memory, which achieves 78% accessing power reduction without performance degradation. The SVSE proposed the new database structure for support vectors and achieves 35% memory reduction in total.

The Retinex preprocessor and SVM processor are fabricated in a 0.13-μm CMOS process as separate ICs and integrated by the NoC off-chip gateway interface. The NoC interface supports the connectivity between two chips. With a demonstration system integrating two chips on an evaluation board, the proposed system achieves 90% recognition rate for a QVGA-sized video images taken in the harsh environment where the illumination condition is dynamically changing. It consumes 92 mW at 1.2 V.

REFERENCES

[1] K. Kim *et al.*, "A 125 GOPS 583 mW network-on-chip based parallel processor with bio-inspired visual attention engine," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 136–147, Jan. 2009.
[2] J.-Y. Kim *et al.*, "A 201.4 GOPS 496 mW real-time multi-object recognition processor with bio-inspired neural perception engine," *IEEE J. Solid-State Circuits*, vol. 45, no. 1, pp. 32–45, Jan. 2010.
[3] S. Lee *et al.*, "A 345 mW heterogeneous many-core processor with an intelligent inference engine for robust object recognition," in *IEEE ISSCC Dig. Tech. Papers*, 2010, pp. 332–333.
[4] R. Bishop, "A survey of intelligent vehicle applications worldwide," in *Proc. IEEE Intell. Veh. Symp.*, 2000, vol. 5, pp. 300–308.
[5] R. Shimizu, "A charge-multiplication CMOS image sensor for low-light-level," in *IEEE ISSCC Dig. Tech. Papers*, 2009, pp. 50–51.
[6] Z. Rahman, G. A. Woodell, and D. Jobson, "A comparison of the Multiscale Retinex with other image enhancement techniques," in *Proc. IS&T's 50th Annu. Conf.: A Celebration of All Imaging*, 1997, pp. 426–431.
[7] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.
[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. J. Comp. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[9] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
[10] J. Platt, "Large margin DAGSVM's for multiclass classification," *Adv. Neural Inf. Process. Sys.*, vol. 12, pp. 547–553, 2003.
[11] G. Madzarov, "A multi-class SVM classifier utilizing binary decision tree," *Informatica*, vol. 33, pp. 233–241, 2009.
[12] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern*, vol. 23, no. 3, pp. 65–685, Mar. 1993.
[13] S. Tan, J. L. Dale, and A. Johnston, "Performance of three recursive algorithms for fast space-variant Gaussian filtering," *Real-Time Imaging*, vol. 9, no. 3, pp. 215–228, 2003.
[14] J. Oh *et al.*, "A 1.2 mW on-line learning mixed mode intelligent inference engine for robust object recognition," in *Proc. Symp. VLSI Circuits*, 2010, pp. 17–18.
[15] J. Oh *et al.*, "A 57 mW embedded mixed-mode neuro-fuzzy accelerator for intelligent multi-core processor," in *IEEE ISSCC Dig. Tech. Papers*, 2011, pp. 130–131.
[16] Y. Ishizuka and Y. Hirai, "Segmentation of road sign symbols using opponent-color filters," in *Proc. 11th World Congress ITS*, Nagoya, Japan, 2004, pp. 18–22.

[17] C. H. Lai and C. C. Yu, "An efficient real-time traffic sign recognition system for intelligent vehicles with smart phones," in *Proc. Int. Conf. Technol. Applic. Artif. Intell.*, Nov. 18–20, 2010, pp. 195–202.
[18] M. Muller *et al.*, "Design of an automotive traffic sign recognition system targeting a multi-core SoC implementation," in *Proc. DATE Conf. Exhib.*, 2010, pp. 532–537.
[19] G. Genov *et al.*, "Kerneltron: Support vector machine in silicon," *IEEE Trans. Neural Network*, vol. 14, no. 5, pp. 1426–1434, Sep. 2003.
[20] S. Chakrabartty *et al.*, "Sub-microwatt analog VLSI trainable pattern classifier," *IEEE J. Solid-State Circuits*, vol. 42, no. 5, pp. 1169–1179, May 2007.
[21] S.–Y. Peng, B. A. Minch, and P. Hasler, "Analog VLSI implementation of support vector machine learning and classification," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2008, pp. 860–863.
[22] K. Kang and T. Shibata, "An on-chip-trainable Gaussian-Kernel analog support vector machine," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2009, pp. 2661–2664.
[23] F. M. Khan *et al.*, "Hardware-based support vector machine classification in logarithmic number systems," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2005, pp. 5154–5157.

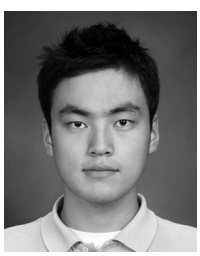


Junyoung Park (S'09) received the B.S. and M.S. degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2009 and 2011, respectively, where he is currently working the Ph.D. degree.

His previous research interests include low-power digital processors for vision applications and a novel architecture for accelerating vision algorithms on a system-on-chip. Recently, he has been investigating efficient implementation of heterogeneous system-on-chip architecture for computer vision processing.

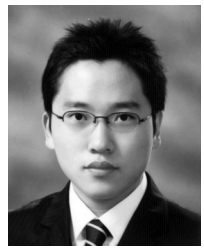
Joonsoo Kwon (S'09) received the B.S. and M.S. degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejon, Korea, in 2009 and 2011, respectively. His M.S. work concerned both low-power processor design and human-like image processing.

He is currently with the Memory Division, Samsung Electronics, where he is a Hardware Designer developing flash memory for low-power and high-speed operation.



Jinwook Oh (S'08) received the B.S. degree from Seoul National University, Seoul, Korea, in 2008, and the M.S. degree from Korea Advanced Institute of Science and Technology (KAIST), Daejon, Korea, in 2010, where he is currently working toward the Ph.D. degree, all in electrical engineering and computer science.

His research interests include low-power digital signal processors for computer vision. Recently, he has been involved with the VLSI implementation of neural networks and fuzzy logics.



Seungjin Lee (S'06–M'12) received the B.S., M.S., and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2006, 2008 and 2011, respectively.

He is currently a Researcher with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. His previous research interests include low-power digital signal processors for digital hearing aids and body-area communication and novel architectures for accelerating neural networks on a system-on-chip. Currently, he is investigating efficient heterogeneous system-on-chip architectures for computer vision processing.

Joo-Young Kim (S'05–M'12) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2005, 2007, and 2010, respectively. His Ph.D. research focused on system-on-a-chip (SoC) architectures and circuit innovations for low-energy object recognition processing.

In 2010, he joined Microsoft Research, Redmond, WA, where he is currently a Research Hardware Design Engineer engaged in the research and development of client and cloud applications which target disruptive end-to-end consumer experiences enabled by heavy use of innovative natural user interfaces and cloud-backed services. More specifically, he is playing a key role in inventing customized architectures and circuits for computationally intensive computer vision algorithms such as camera tracking, dense 3-D reconstruction, motion recognition, and scene adaptation at power budgeted mobile devices. He has authored and coauthored over 35 journal publications and conference presentations in the area of solid-state circuits, including the IEEE International Solid-State Circuits Conference, Symposium on VLSI Circuits, and the IEEE JOURNAL OF SOLID-STATE CIRCUITS. He has been invited to talk his researches on computer vision and SoC validation at both industrial and research organizations such as Intel, nVidia, Texas Instruments and IMEC.

Dr. Kim won two ISSCC presentations and two ISSCC/DAC design contests from 2008 through 2010 for three parallel SoCs architected and designed by him.

Hoi-Jun Yoo (M'95–SM'04–F'08) received the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea.

He was the VCSEL pioneer in Bell Communications Research, Red Bank, NJ, and Manager of the DRAM Design Group, Hyundai Electronics, designing from 1 M DRAM to 256 M SDRAM.

Currently, he is a full professor of Department of Electrical Engineering at KAIST and the director of the System Design Innovation and Application Research Center (SDIA). From 2003 to 2005, he served as the full time Advisor to the Minister of Korean Ministry of Information and Communication for SoC and Next Generation Computing. His current research interests are Bio Inspired IC Design, Network on a Chip, Multimedia SoC design, Wearable Healthcare Systems, and high speed and low power memory. He has published more than 250 papers, and wrote or edited 5 books, "DRAM Design" (1997, Hongneung), "High Performance DRAM" (1999 Hongneung), "Low Power NoC for High Performance SoC Design" (2008, CRC), "Mobile 3-D Graphics SoC" (2010, Wiley), and "BioMedical CMOS ICs" (Co-editing with Chris Van Hoof, 2010, Springer), and many chapters of books.

Dr. Yoo received the Korean National Medal for his contribution to Korean DRAM Industry in 2011, the Electronic Industrial Association of Korea Award for his contribution to DRAM technology the 1994, Hynix Development Award in 1995, the Korea Semiconductor Industry Association Award in 2002, Best Research of KAIST Award in 2007, Design Award of 2001 ASP-DAC, Outstanding Design Awards of 2005, 2006, 2007, 2010, 2011 A-SSCC, and Korean Scientist of the Month Award (Dec. 2010). He is a member of the executive committee of Symposium on VLSI, and A-SSCC. He was the TPC chair of the A-SSCC 2008, a guest editor of IEEE JSSC and IEEE T-BioCAS. He was the TPC Chair of ISWC (International Symposium on Wearable Computer) 2010, IEEE Fellow, IEEE Distinguished Lecturer ('10–'11), Far East Chair of ISSCC ('10–'11), and currently ISSCC Technology Direction Sub-committee Chair and an associate editor of IEEE TCAS-II.