Low-Power Scalable 3-D Face Frontalization Processor for CNN-Based Face Recognition in Mobile Devices

Sanghoon Kang, *Student Member, IEEE*, Jinmook Lee, *Student Member, IEEE*, Kyeongryeol Bong, *Student Member, IEEE*, Changhyeon Kim, *Student Member, IEEE*, Youchang Kim, *Student Member, IEEE*, and Hoi-Jun Yoo, *Fellow, IEEE*

***Abstract*— A low-power scalable 3-D face frontalization processor is proposed for accurate face recognition in mobile devices. In spite of recent improvement in face recognition accuracy mainly from convolutional neural networks (CNNs), their performance is limited to face images with frontal view. For face recognition with human-level accuracy in real-life environment, in which most of the face images are captured from arbitrary angles, 3-D face frontalization is essential as a preprocessing stage for CNN-based face recognition algorithms. The proposed face frontalization processor shows scalability in two aspects: image resolution and accuracy. For low-power consumption and scalability, the processor proposes three features: 1) scalable processing element (PE) architecture with workload adaptation; 2) accuracy scalable regression weight quantization to reduce the external memory access (EMA) down to 81.3%; and 3) pipelined memory-level zero-skipping to further reduce the EMA by 98.4% without any latency overhead. From the proposed EMA reduction features, the EMA is reduced by 99.7% with little accuracy degradation in face frontalization results. The proposed face frontalization processor is implemented in 65-nm CMOS process, and it shows 4.73 frames/s) throughput. Moreover, power consumption of the implemented face frontalization processor is 0.53 mW, which is suitable for applications on mobile devices.**

***Index Terms*— Accuracy scalability, convolutional neural network, face recognition, low-power processor, 3-D face frontalization.**

I. Introduction

RECENTLY, implementation of face recognition technology in mobile devices has been widely investigated for user authentication [1], and also for recognition of others in social interactions [2]. The main factor for the recent interest in face recognition is the dramatic increase of its recognition accuracy, driven by deep convolutional neural networks (CNNs). Face recognition algorithms based on CNN enabled the computers to recognize people's face more accurate compared to human's ability [3].

However, its performance is limited to the aligned face images. When face images are not captured from the front but from arbitrary angles, such as examples in the LFW dataset [4], the accuracy of CNN-based face recognition is severely degraded than that of human's [5]. Therefore, aligning face images 3-dimensionally to a frontal view, i.e. 3-D face frontalization is essential in the face recognition system to achieve human-level recognition accuracy.

In addition to the algorithmic improvements of the face recognition systems with CNNs, there has been a breakthrough in hardware's perspective. A dedicated ultra-low-power CNN processor [6] targeting face recognition has been introduced to realize computations and data intensive CNN operations under 1 mW power consumption for battery-powered mobile devices.

However, without dedicated hardware for the 3-D face frontalization, the process must take place on mobile application processor (AP), to ensure high accuracy of the face recognition result. However, processing the 3-D face frontalization algorithm on a commercial mobile AP consumes much energy. Therefore, it forbids us from long battery life even with the dedicated low-power CNN processor. Tested on Samsung's Exynos 5422 [7], the 3-D face frontalization algorithm runs at a 3.63 frames-per-second (fps) throughput, while consuming an average of 4.6 W.

Furthermore, CNN-based face recognition algorithms vary widely by their input image resolution, based on their target application. While networks targeting face recognition in low-quality surveillance footage takes in low-resolution face images as small as 6×6 pixels per image [8], [9], others target much higher resolution face images up to 220×220 pixels per image [10]. Thus, a scalable architecture, which can support various image resolutions, is important not only for the CNN accelerator but also for the face frontalization hardware, which functions as a preprocessor in the face recognition system.

In this paper, we present a low-power scalable 3-D face frontalization processor for accurate face recognition in mobile devices. The proposed processor is scalable in 2 different aspects: image resolution, and frontalization accuracy. To achieve low-power consumption with scalability, 3 key

Manuscript received January 7, 2018; revised May 11, 2018; accepted June 1, 2018. Date of publication June 8, 2018; date of current version December 11, 2018. This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), Ministry of Science, ICT & Future Planning, under Grant NRF-2015R1A2A1A05001889. This paper was recommended by Guest Editor A. Marongiu. *(Corresponding author: Sanghoon Kang.)*

The authors are with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea (e-mail: sanghoon_kang@kaist.ac.kr).

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/JETCAS.2018.2845663

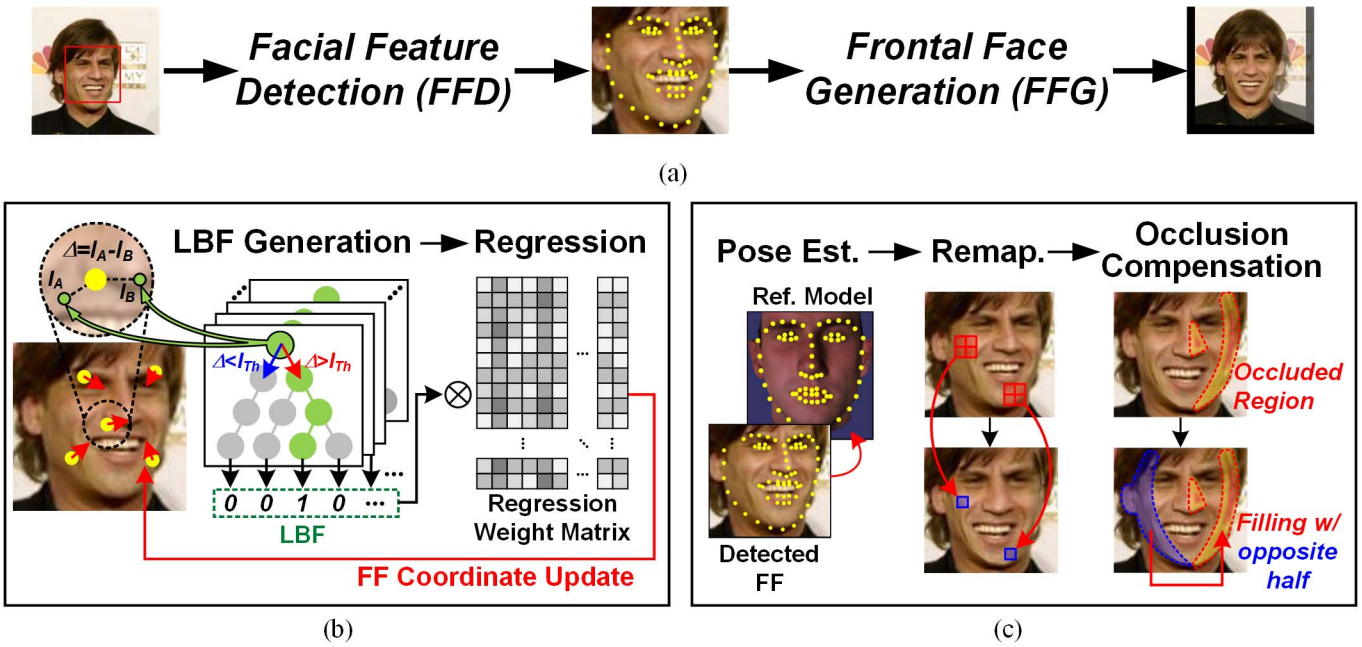


Fig. 1. 3-D face frontalization algorithm: (a) overall flow; (b) facial feature detection stage; (c) frontal face generation stage.

features are proposed. First, a scalable processing engine (PE) architecture is proposed with its workload adaptation to deal with varying computation requirement. Second, we propose an accuracy scalable weight quantization scheme with K-means clustering which could greatly reduce the external memory access (EMA). At last, a zero skipping from the memory-level is proposed with tree-level pipelining to prevent fetching of unnecessary data from external memory and to reduce the overall latency.

The rest of this paper is organized as follows. Section II explains about previous works related to this paper. In Section III, the 3-D face frontalization algorithm will be described with its challenges as a mobile application. Section IV discusses the overall processor architecture with 3 key features for low-power processing. The implementation results of the proposed processor will be shown in Section V, followed by the conclusion in Section VI.

II. Related Work

With the rise of CNN, deep learning based face recognition algorithms have reached, or surpassed the ability of human's visual system. As first stated in [5], the face frontalization remains as a necessary pre-processing stage, and adopting frontalization in front of CNN has become a norm in the field of face recognition. Nonetheless, no dedicated hardware for the face frontalization has been proposed, nor implementation of the algorithm on mobile/embedded environment has been introduced. This paper is the first to introduce a hardware implementation of the face recognition algorithm. For that reason, this paper compares the designed processor's performance with the baseline of software implementation on the mobile application processor.

The 3-dimensional face frontalization algorithm adopted in this paper is [11]. It showed an improved performance over [5] using symmetry based occlusion compensation. To estimate a transformation matrix from the input image into a frontalized face image, location of facial features should be extracted. Reference [11] adopted the supervised descent method (SDM) [12] to estimate the coordinate of feature points. SDM is widely adopted facial feature detection algorithm in both academia and commercial applications, therefore is selected as the accuracy baseline of the feature detection algorithm.

The facial feature detection algorithm [13], which is adopted in this paper, extracts local binary features based on tree search and estimates feature coordinates based on regression. It outperformed SDM in both accuracy and speed. However, [13] requires excessive EMA, which leads to large energy consumption. This paper aims to minimize the EMA with minimal accuracy loss through the proposed features. More detailed explanation of the adopted frontalization algorithm [11], [13] will be given in the next Section.

III. 3-D Face Frontalization Algorithm

The overall flow of the 3-D face frontalization algorithm is shown in Fig. 1. Briefly, the algorithm is divided into two stages: 1) facial feature detection (FFD) stage [13] and 2) frontal face generation (FFG) stage [11]. The FFD stage finds facial feature points, or facial landmarks, from the given input face image. Based on the detected facial features, frontal face image is generated to match the pose of the input face image with that of a predefined reference model. The detailed explanation of each stage will be given followed by the challenges of the 3-D face frontalization algorithm in implementing on mobile devices.

A. Facial Feature Detection (FFD) Stage

The facial features are localized through iterative regression of the local binary features (LBFs), as shown in Fig. 1(b). The FFD starts with a naïve prediction of the facial landmarks'

positions S^0 based on the mean values of the landmark coordinates from the training dataset as shown in (1), where x, y are the coordinates of facial landmarks. Every t^{th} iteration of LBF regression calculates the coordinate increment ΔS^t, based on which the facial landmark coordinates are updated every iteration as shown in (2):

$$S^0 = [x_1^0, y_1^0, \ldots, x_N^0, y_N^0], \tag{1}$$

$$S^t = S^{t-1} + \Delta S^t. \tag{2}$$

Multiple search trees are allocated for each feature point, and the result of each tree is a one-hot binary vector. Each node in the search tree is evaluated by comparing the intensity difference ΔI with the threshold I_{th}. The intensity difference is calculated between the two sampled pixels:

$$\Delta I = I_A - I_B, \tag{3}$$

where the coordinates of the sampled pixels A and B are expressed in relative coordinates with its origin placed at the predicted landmark from the previous iteration. After the evaluation of the node, it propagates to the next leaf node, and continues until it reaches the bottom of the tree. With the tree depth of D, 2^{D-1} dimension binary vector ϕ_l^t is generated in one-hot format highlighting the selected bottom leaf node from the unselected nodes. The resulting LBF Φ^t is the concatenation of all the vectors from the search trees:

$$\Phi^t = [\phi_1^t, \phi_2^t, \ldots, \phi_L^t], \tag{4}$$

After generating the LBF through the evaluation of all the trees, the LBF is then multiplied with a previously trained regression weight matrix W^t. From the multiplication result, the increments of the facial landmark coordinates are acquired:

$$\Delta S^t = \Phi^t \cdot W^t. \tag{5}$$

From (2) and (5), facial feature coordinates S^t are updated at the end of every iteration. Final decision of the facial feature coordinates S^T are extracted after a total of T iterations.

B. Frontal Face Generation (FFG) Stage

The facial feature points, which are detected from the previous FFD stage, are utilized as landmarks to estimate the pose of the given face relative to the reference face model as shown in Fig. 1(c). Given the coordinates of the landmarks on the 3-D reference model $P_n = (X, Y, Z)^T$, and the coordinates of the corresponding landmarks on the input face image $p_n = (x, y)^T$, the projection matrix C_Q is estimated:

$$p_n \sim C_Q \cdot P_n \tag{6}$$

for all $n (1 \leq n \leq N)$ [11].

Based on the projection matrix computed from (6), frontal face view is generated by remapping and interpolating pixels from the input face image. If a part of the face is occluded in the input image due to a pose of the face, a smeared region will appear in the generated frontal face image due to lack of pixel data of the occluded region. For these occluded regions highlighted in yellow as in Fig. 1(c), intrinsic left-right symmetry characteristic of human face can be utilized. Through filling the occluded regions with the pixel values from the opposite half of the face, occluded regions are compensated.

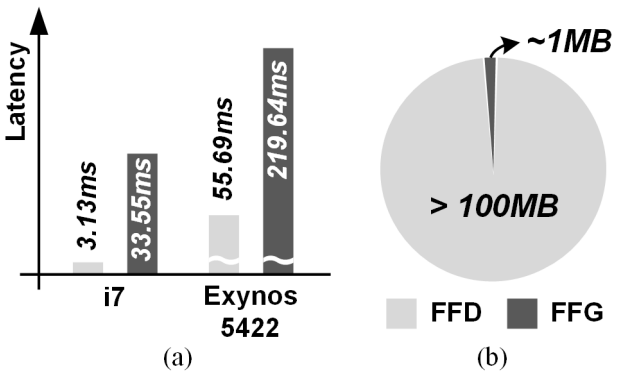


Fig. 2. Process analysis of 3-D face frontalization algorithm: (a) latency; (b) external memory access (EMA).

C. Design Challenges

Implementing the 3-D face frontalization algorithm on general-purpose mobile processors consume much energy due to the large amount required EMA. In addition, highly unbalanced workload between the stages of 3-D face frontalization algorithm makes it inefficient to accelerate by allocating identical processing units. Samsung Exynos 5422 [7] consumes 4.6W while processing 250 × 250 image frames at 3.63 fps, and it will last less than 3 hours using a standard smartphone battery. To alleviate the problem of such high power consumption, this paper present the low-power scalable 3-D face frontalization processor.

As mentioned, there are two main design challenges in implementing low-power face frontalization processor. First, the two stages of the frontalization algorithm, the FFD and the FFG, show highly unbalanced workload. As shown in Fig. 2(a), the FFG stage takes 5x∼10x more computation time compared to the FFD stage. While most of the fundamental operations required in both the FFD stage and the FFG stage are identical; the matrix multiplication, the different characteristic of the data-centric FFD stage, and the computation-centric FFG stage makes it inefficient to be implement on a single processor.

Secondly, but more importantly, the regression of features in the FFD stage requires excessive amount of external memory access (EMA). It is mainly due to large regression weight matrix shown in Fig. 1(b). This leads to the EMA over 100 MB per single frame, as shown in Fig. 2(b). The EMA is the main factor for large energy consumption, which should be taken into consideration prior to the energy used in the computation itself. Rough energy cost of basic operations and memory fetch is reported in [14], and memory fetch consumes 350x ∼ 1400x more energy compared to multiply or add operations. 8.02 MOP (Mega Operations) are required for 250 × 250 image frontalization, and based on the energy estimation of [14], the EMA consumes 1000x more energy compared to the computation operations. Therefore, it is essential to decrease the EMA for 3-D face frontalization with low power consumption.

This paper proposes three key features to alleviate the introduced challenges. The scalable 3-D face frontalization processor architecture will be proposed to accelerate both

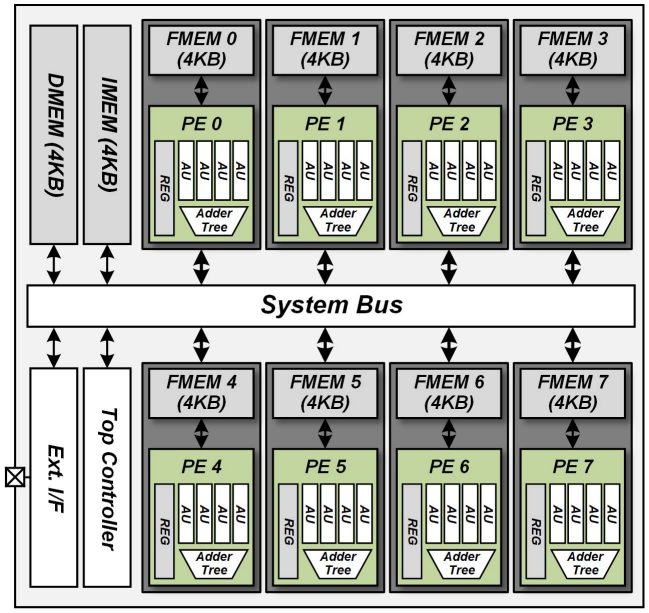


Fig. 3. Overall architecture of proposed 3-D face frontalization processor.

the FFD stage and the FFG stage on the single processor. Moreover, a workload adaptation is proposed for energy-efficient implementation of the unbalanced computation requirements of the FFD stage and the FFG stage. In addition, to decrease the EMA in the FFD stage, the paper proposes the accuracy scalable regression weight quantization using K-means clustering. Finally, to further reduce the EMA, pipelined memory-level zero-skipping of the weight matrix is proposed.

IV. Proposed 3-D Face Frontalization Processor

A. Overall Processor Architecture

The overall architecture of the proposed face frontalization processor is shown in Fig. 3. It mainly consists of 8 identical processing element (PE) and 4KB frontalization memory (FMEM) for storing the input or output data from local PEs. In addition, a top-level RISC controller with 4KB instruction memory (IMEM) and 4KB data memory (DMEM) is incorporated in the proposed processor.

Each of the FMEM is attached to each PE, and functions as a local scratchpad memory, i.e. data from the FMEM can only be accessed from the corresponding PE in the same pair. The FMEM stores the input operands for the PE, and stores the output value of the arithmetic operations computed inside the PE. Each PE-FMEM pair operates independently, and there is no data transaction between the pairs.

The PE-FMEM pairs can only communicate with the top-level controller and its corresponding data memory, which is the DMEM. Data required from multiple PEs is stored inside the DMEM and shared across the PEs. Sharing a top-level data memory for data, which is used across all the PEs, enables avoiding the data overlap among the local memories in different PEs. In addition, dedicating the FMEM to data utilized and generated in each PE helps enhancing the local memory access speed in the PE.

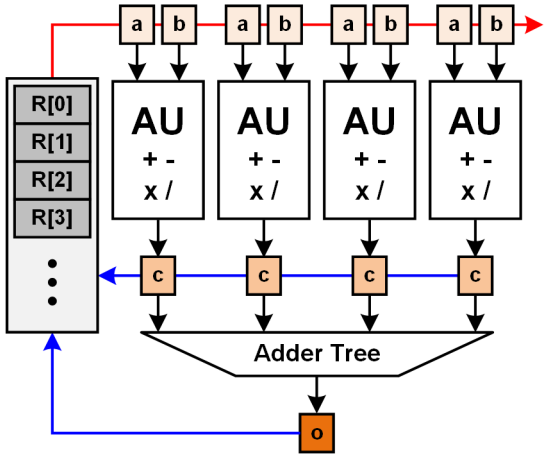


Fig. 4. Processing element (PE) architecture of the proposed face frontalization processor.

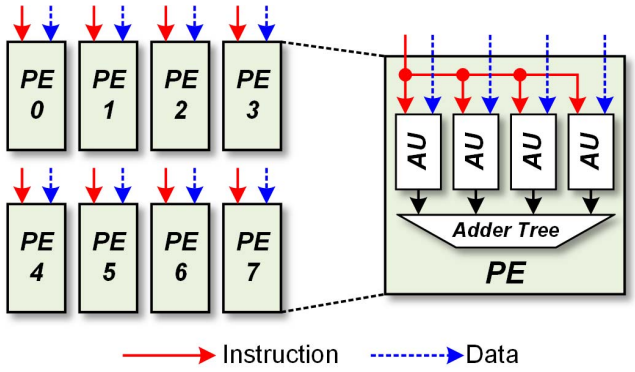


Fig. 5. Hierarchical processing architecture for scalability: inter-PE-level, and intra-PE-level.

B. Scalable PE Architecture

Fig. 4 shows the architecture of the PE. It consists of 4-way parallel arithmetic units (AU), 4-way adder tree, the register file, input register for the AU operands, and the output register. The register file holds temporal data such as the AU operands, and the computation results. Moreover, the register file can fetch data from the FMEM, which is paired with its PE, or from the DMEM, which is shared globally.

Each AU consists of an adder/subtractor, a multiplier, and a divider. All the AUs are of the same configuration, each incorporating 24-bit fixed-point arithmetic units for each fundamental arithmetic operation. However, the AUs share a single divider for area efficiency. By using a single 3-stage pipelined divider across the 4 AUs rather than implementing separate dividers for each of the AU, the area required for divider is decreased by 4x. By sharing the divider, computation cycles for dividing operation increase twice, but it barely affect the overall throughput since there is very little dividing operations required for 3-D face frontalization algorithm.

The results of 4-way parallel AUs are aggregated using the 4-way adder tree to efficiently accelerate the transformation of the 3-D homogenous coordinates, which accounts for the majority of the computation required in the 3-D face frontalization algorithm. The homogenous coordinate of the point in 3-dimensional space $X = (x, y, z)^T$ is represented as

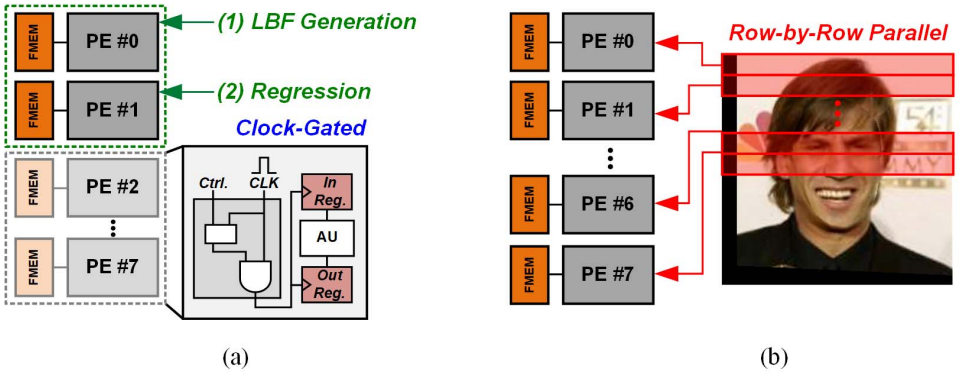


Fig. 6. Workload adaptation through PE allocation: (a) FFD stage; (b) FFG stage.

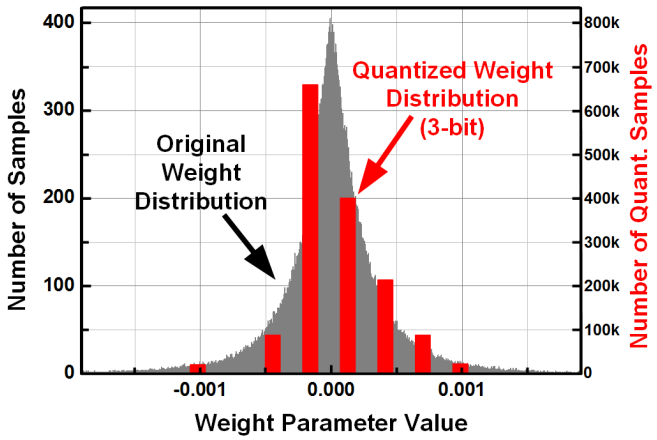


Fig. 7. Original weight distribution (gray), and 3-bit quantized weight distribution (red).

$X_h = (x, y, z, 1)^T$. The coordinate projection of the given homogenous coordinate X_h is through multiplication with 3×4 projection matrix M:

$$\begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} = M \cdot X^T = \begin{pmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \tag{7}$$

Each of the output coordinate element u', v', and w' is acquired from 4 multiplications followed by adding up the results of the multiplication. This can be directly mapped to the PE architecture of 4-way AUs with 4-way adder tree.

The proposed face frontalization processor is designed in hierarchical processing architecture as shown in Fig. 5. The architecture of the PEs on the inside, i.e. in intra-PE-level, is a single instruction stream, multiple data stream (SIMD) architecture. By making all the AUs to go through the same arithmetic operations with different operands, the PEs can efficiently accelerate the required operations such as 3-D coordinate projection as (7).

In the inter-PE-level, the processor functions as a multiple instruction stream, multiple data stream (MIMD) architecture, so that different PEs can be assigned to different tasks. With the MIMD architecture between the PEs, it is also possible to control the PEs to operate homogeneously by feeding identical instruction to all the PEs. The hierarchical architecture enables scaling the processor depending on the resolution of the input

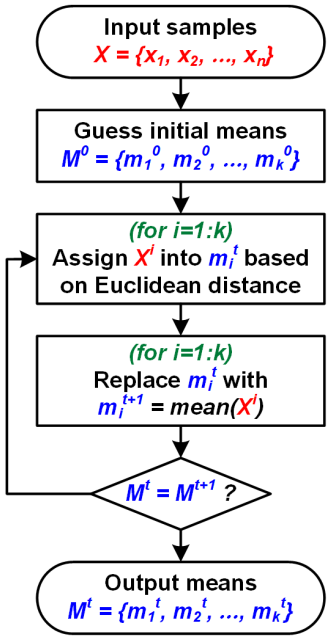


Fig. 8. K-means clustering algorithm flow diagram.

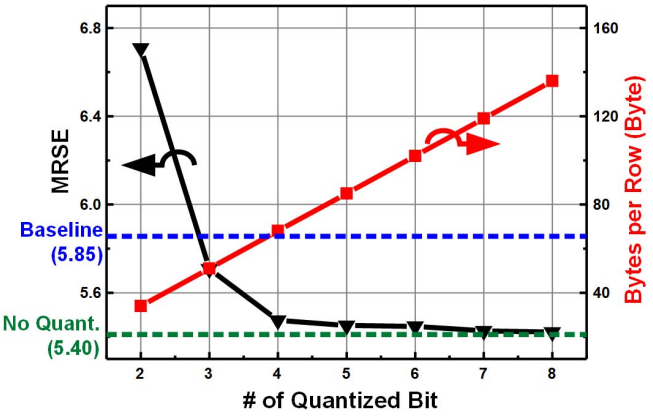


Fig. 9. Accuracy scalable weight quantization by K-means clustering: accuracy comparison with baseline method, and required bytes per row for representation of regression matrix.

image on which the face recognition system targets. For larger image resolution, larger number of PEs can be integrated to perform face frontalization.

C. PE Workload Adaptation

As mentioned in Section II, the FFD stage and the FFG stage of the frontalization algorithm show largely unbalanced workload. The FFD stage is parameter dominant, i.e. it requires a lot of EMA, but does not require much computation, and the FFG stage is computation dominant.

While processing the computation dominant FFG stage, all the PEs are allocated to accelerate the process in parallel. The frontalized face image is generated from remapping the pixels from original face image. Each row of the frontalized face image is allocated to each PE. Since the rows of the generated frontal face image have no coherency between each other, the rows can be directly mapped to different PEs as shown in Fig. 6. Moreover, the occlusion compensation of the FFG stage utilizes the intrinsic left-right symmetry of human face, so data dependency rises only inside each row. Because of the FMEM local memory architecture and row-by-row parallel workload assignment, data dependency is solved without any data transaction between the PEs.

Due to the workload difference, different number of PEs are allocated for the FFD stage and the FFG stage as shown in Fig. 6. For the FFD stage, which does not require much computation, only two PEs are activated: one for LBF generation through binary tree search, and the other for the regression. The remaining PEs are not utilized in the FFD stages, and are clock-gated to reduce dynamic power, resulting 37.14% reduction in power consumption.

The proposed face frontalization processor targets its existence as a co-processor with the CNN accelerator, and its function is not limited to preprocessing the input image. Between the layers of CNN, 1-dimensional vector processing is required for batch normalization [15]. Running it on 2-dimensional multiply-and-accumulate (MAC) arrays of CNN accelerator [6], [13] is highly inefficient due to the 1-dimensional nature of batch normalization. Integration of the proposed frontalization processor with the CNN accelerator can benefit not only from the increased accuracy of face recognition result compared to the recognition system with no face frontalization, but also from increased efficiency in CNN processing.

D. Accuracy Scalable Regression Weight Quantization through K-means Clustering

The large EMA of the FFD stage is a major reason for high power consumption of the face frontalization algorithm. To alleviate this problem, this paper proposes an accuracy scalable quantization method of the regression weight matrix, which is over 100 MB per frame. For the full process of the regression, five of $(1200\times64)\times(68\times2)$ size matrices are required (1200: number of trees, 64: number of final leaf node of each tree, 68: number of feature points, 2: 2-dimensional coordinate space).

The gray-colored graph on Fig. 7 shows the original distribution of elements in regression weight matrix. The distribution is highly concentrated near zero. By clustering the original distribution into small number of clusters, it is possible to use less number of bits to represent the given data approximately. However, approximation of the learned weight matrix necessarily causes degradation in algorithm accuracy. The lesser bits used to represent original data, the more degradation in accuracy the algorithm shows. This work proposes weight quantization with K-means clustering as shown in Fig. 8, to reduce the number of bits required for weight expression with minimum accuracy degradation.

As shown in Fig. 9, the EMA required to represent the regression weight matrix decreases linearly as the number of quantization bit decreases. At the same time, the accuracy of FFD is decreased. Therefore, the burden of external memory access can be alleviated with scaling accuracy. The FFD error is represented in mean root square error (MRSE), which is calculated as the average of root square distance of the estimated feature points to the ground truth points, and then normalized to pupil-to-pupil distance. In this work, the original data is clustered down to $2^3 = 8$ elements to minimize the EMA required for the regression weight matrix while maintaining the higher accuracy compared to the FFD baseline of [11], which uses SDM [12]. Clustering the whole regression

TABLE I
FACIAL FEATURE DETECTION ERROR WITH REGRESSED WEIGHT MATRIX

Quantization Method	Mean Root Square Error (MRSE) Number of Quantized Bits						
	8-bit	7-bit	6-bit	5-bit	4-bit	**3-bit**	2-bit
Uniform Distribution	5.42	5.43	5.47	5.56	5.87	8.94	46.72
Gaussian Distribution	5.89	5.98	6.17	6.4	6.76	7.37	8.73
Expectation Maximization	5.95	6.04	6.53	8.9	13.3	20.17	24.47
K-Medoids Clustering	5.42	5.43	5.41	5.43	5.52	5.76	6.76
K-Means Clustering	5.42	5.42	5.45	5.45	5.47	**5.71**	6.71

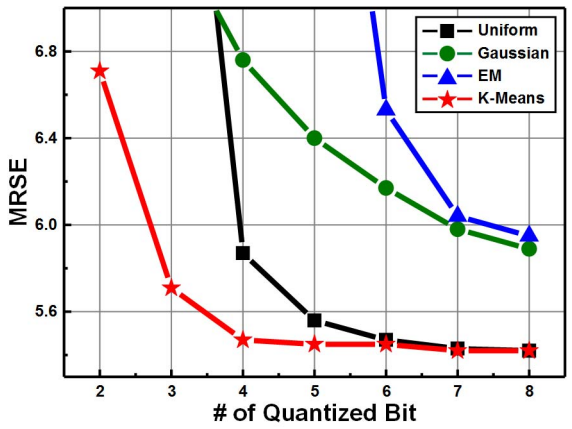


Fig. 10. Comparison of clustering algorithms over mean root square error (MRSE) of facial feature detection (FFD).

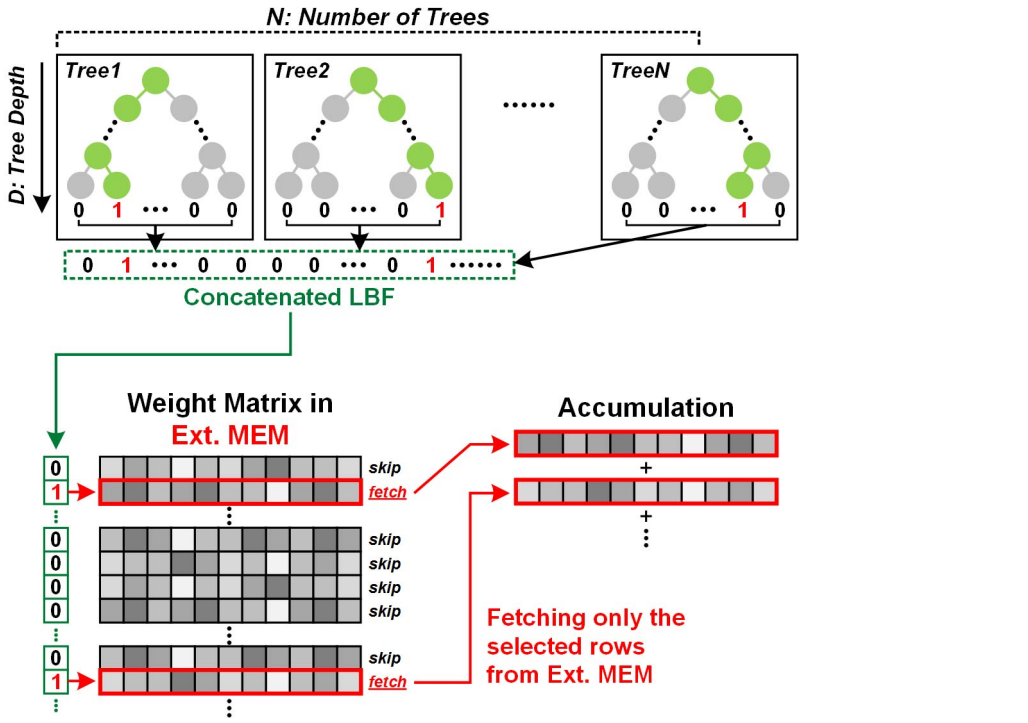


Fig. 11. Memory-level zero-skipping of the regression weight matrix.

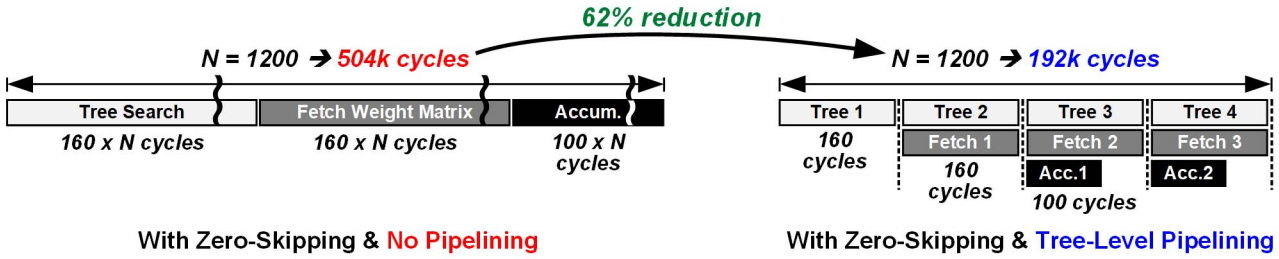


Fig. 12. Tree-level pipelining of the facial feature detection (FFD) stage.

weight matrix into 8 points makes it possible to represent every element of the matrix using only 3-bit each.

Table 1, and Fig. 10 shows the comparison of FFD accuracy over 5 different clustering algorithms: uniform distribution-based clustering, Gaussian distribution-based clustering, expectation maximization (EM) clustering [16], K-medoids clustering [17], and K-means clustering [18]. The accuracy of the clustering algorithms are evaluated and are compared using Helen dataset [19].

As shown in both Table 1 and Fig. 10, K-means clustering showed the best MRSE performance over all the other clustering schemes. The uniform distribution-based clustering showed fair performance when using larger number of bits, but it collapsed dramatically as the number of bits decreases. The EM clustering showed the worst performance, followed by the Gaussian distribution-based clustering, probably due to its original data distribution being far from the Gaussian distribution itself. The K-medoids clustering scheme showed almost similar results compared to the K-means clustering scheme, but the K-means was a little bit better when it comes to extremely small number of quantization bits.

Given that the K-means clustering algorithm showed best performance over other clustering schemes, the minimum number of quantization bits is decided based on the comparison with the baseline result from SDM FFD algorithm. As shown in Fig. 9, FFD based on the introduced local binary feature regression algorithm shows better performance over SDM even if the regression weight matrix is quantized. The clustered data using K-means clustering showed better performance up to 3-bit quantization over SDM. However, as shown in the graph, 2-bit quantization showed dramatic degradation in FFD accuracy.

In summary, accuracy scalable quantization of regression weight matrix using K-means clustering is implemented to reduce EMA with scaling accuracy of FFD. Through 3-bit quantization with the proposed scheme, the previous 100 MB EMA is reduced by 81.25% down to 18.75 MB. Meanwhile, the accuracy degradation of the FFD is negligible, showing 5.64% MRSE increase compared to the previous FFD algorithm [13]. Moreover, the frontalization results show little difference after the 3-bit quantization the FFD weights, and the results will be visualized in the following section.

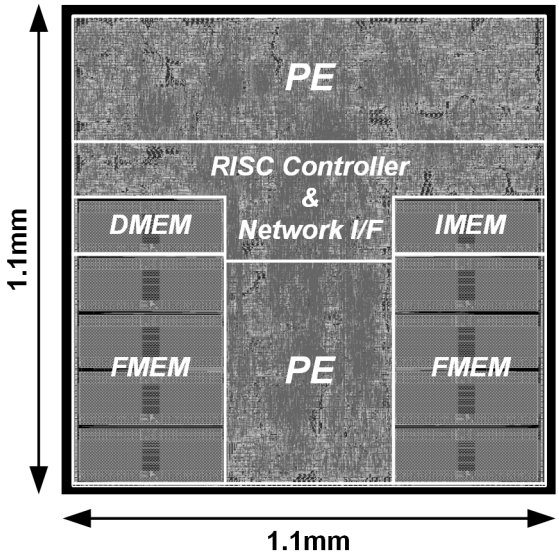


Fig. 13. Layout photograph.

TABLE II
CHIP SPECIFICATIONS

	Specifications	
Technology	Samsung 65 nm Logic CMOS	
Area	1.10 mm x 1.10 mm	
Supply Voltage	1.2 V	
Frequency	5 MHz ~ 100 MHz	
Framerate	5 MHz	4.73 fps
	100 MHz	94.60 fps
Facial Features	68	
Resolution	250 x 250	
Power	5 MHz	0.53 mW
	100 MHz	4.09 mW

E. Pipelined Memory-Level Zero-Skipping Regression

For further reduction of the EMA in the FFD stage, this paper propose memory-level zero-skipping of regression weight matrix. This feature utilizes the inherent characteristic of local binary features (LBF) used throughout the FFD stage. As briefly explained in the previous section, the FFD stage is the iteration of two steps: LBF generation, and regression. LBF is generated from binary tree search, and then the generated LBF is multiplied with the regression weight matrix.

As shown in Fig. 11, LBF is generated from the concatenation of the vectors resulting from the tree search. Moreover, only one of the element from the vector is activated; i.e. all the other elements are zero except one activated element. Therefore, with tree depth $D = 7$, 63 elements out of 64-long tree search vector is zero. The generated LBF from concatenation of the vectors will be multiplied with the regression weight matrix, which is fetched from external memory into the local FMEM.

Because most of the elements of LBF is zero, the multiplication result of zero with the corresponding row from the regression matrix is zero. Therefore, there is no need to fetch all the rows of the weight matrix from external memory. As shown in Fig. 11, through fetching only the selected rows from the LBF result, it is possible to decrease the EMA. Moreover, due to the remaining elements of LBF being one, the selected rows are accumulated; i.e. there is no need for multiplication at all for regression. All in all, the essential rows of the regression weight matrix is selected through evaluation of LBF, and only the selected rows are fetched into the local FMEM, while skipping other rows at the memory-level.

This feature can decrease the power consumption not only from substituting large size matrix multiplication into accumulation of selected matrix rows, but also from dramatic decrease in EMA induced from memory-level zero-skipping. From the

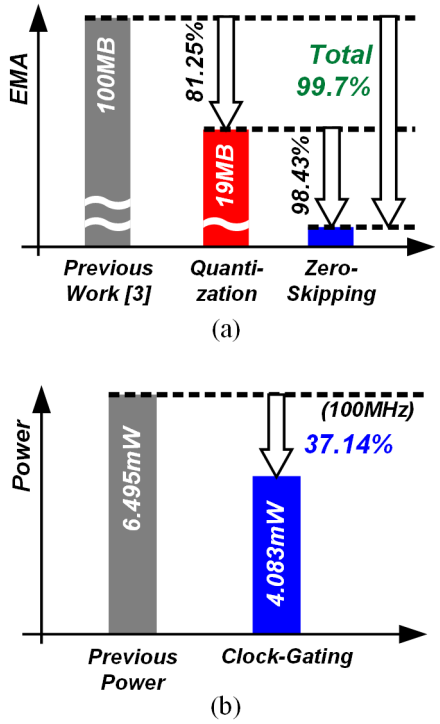


Fig. 14. Implementation results: (a) EMA reduction through quantization and memory-level zero-skipping; (b) Power reduction through workload adaptation and clock-gating.

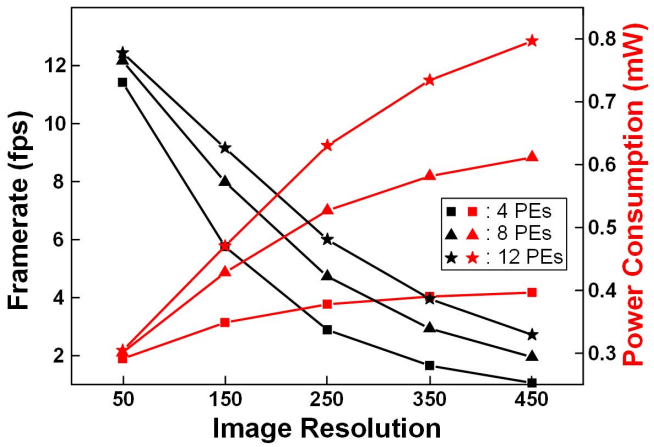


Fig. 15. Resolution scalability: Framerate and power consumption change with respect to image resolution for different number of PEs.

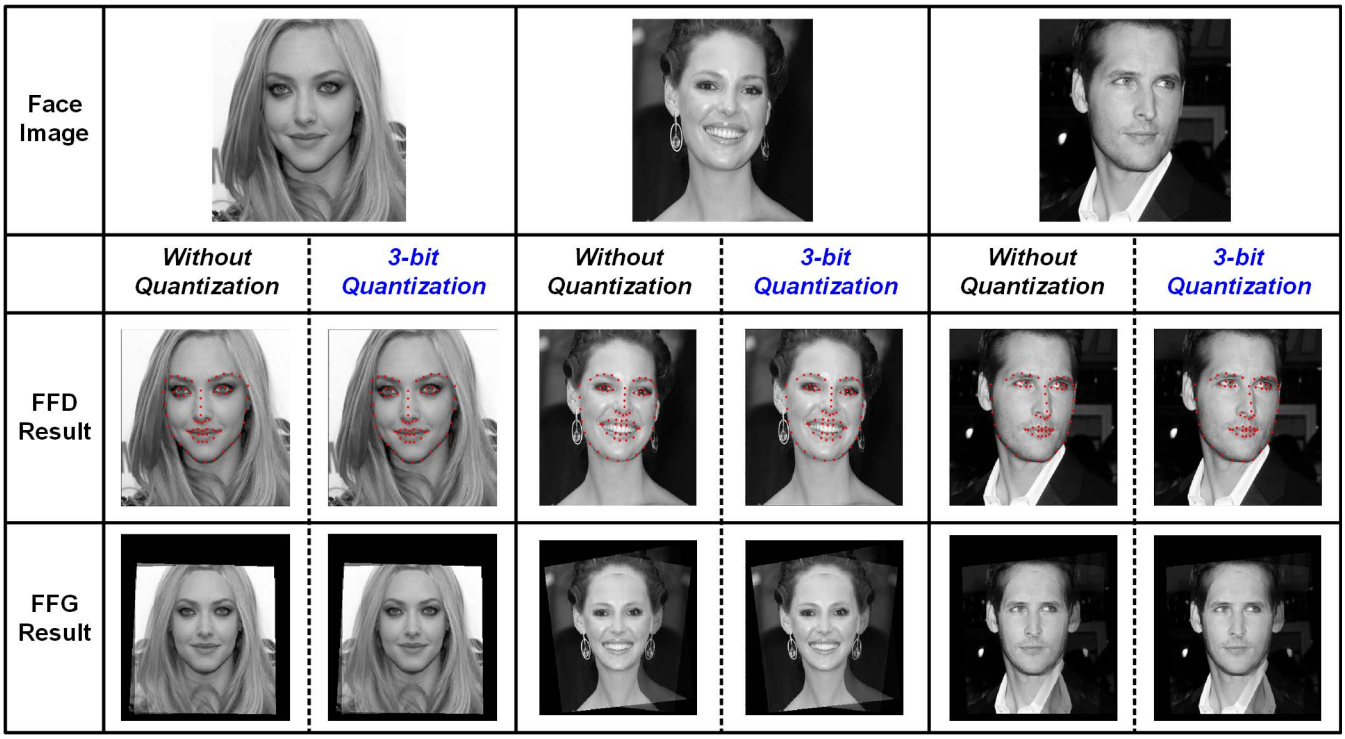


Fig. 16. Facial feature detection (FFD) and frontal face generation (FFG) result with and without regression weight quantization.

19 MB of EMA, which is already a reduced EMA from the proposed 3-bit quantization, the EMA of the FFD stage is further clipped down by 98.4% through implementing the zero-skipping.

Memory-level zero-skipping forces the processor to wait for the evaluation of LBF to find out which of the rows should be fetched. This characteristic forbids hiding the memory access latency as shown in Fig. 12. With the number of trees $N = 1200$, it takes 504k cycles for the FFD stage. However, instead of waiting for the entire LBF to be evaluated, we tiled the LBF evaluation steps in tree-level. After the evaluation of each tree, 64-long one-hot vector, which is a part of the LBF, is generated. The resulting vector indexes a single row out of N rows, which are to be accumulated. By tiling tree search, fetch, and accumulation steps in tree-by-tree units, it is possible to pipeline the FFD stage as shown in Fig. 12. Tree-level pipelining enables hiding the latency of weight fetching under the next tree evaluation step. Despite the implementation of a pipeline, which usually leads to an overhead from pipeline registers, no pipeline register overhead has occurred. Because different PEs are allocated for the tree search and the regression operation as shown in Fig. 6, it is able to reuse the resources that are already integrated for the parallelism of the FFG stage. Thanks to the proposed zero-skipping and tree-level pipelining, latency of the FFD stage is reduced by 61.5%.

V. Implementation Results

A. Layout Implementation Results

The proposed processor is implemented and simulated in Samsung 65nm CMOS logic process. Fig. 13 shows the layout photograph of the implemented chip. The chip occupies 1.1mm $\times$ 1.1mm area. 8 PEs each with their local memory FMEM, and top controller with global memories IMEM, DMEM are implemented on a single die.

Table II is the performance summary of the implemented chip. The processor is operating at 1.2V supply voltage, in the clock frequency range of 5 MHz to 150 MHz. The implemented face frontalization processor detects 68 facial feature points to analyze the input face image with respect to the reference face model, and generates the frontalized face image whose resolution is 250 $\times$ 250.

Operating in 100 MHz clock frequency, implemented 3-D face frontalization processor shows 94.6 fps while consuming 4.09 mW of power. If the operating frequency is lowered down to 5 MHz, the framerate of the frontalization is 4.73 fps, with 0.53 mW power consumption. Achieving such low-power consuming processor for 3-D face frontalization makes it possible to implement an always-on face recognition system in battery-powered mobile device.

B. Simulation Results

Fig. 14(a) shows the effectiveness of the proposed schemes i.e. regression weight quantization and memory-level zero-skipping, in terms of EMA. The original EMA of the previous FFD stage is 100 MB per frame, and the EMA is reduced over 99% by the two EMA reduction schemes. First, quantizing the regression weight matrix through K-means clustering reduce the EMA by 81.25%. Second, memory-level zero-skipping enables selecting only the necessary rows of the regression weight matrix located in external memory, thus reducing the EMA 98.43% more. Therefore, 99.7% EMA reduction is achieved in the FFD stage and the total EMA that is required to process a single frame is 4.37 MB.

The proposed face frontalization processor is .scalable with respect to different image resolutions. In addition, higher framerate can be achieved by integrating additional PEs in the core. Fig. 15 shows the framerate and the power consumption of the frontalization processor in different image resolutions, running at 5 MHz clock frequency. The 3 different shaped curves indicate different number of PEs integrated in the processor. When high-resolution image is required, faster framerate could be achieved through integrating more PEs, with the sacrifice of power consumption.

Proposed 3-bit weight quantization through K-means clustering reduce a great amount EMA, in sacrifice of little FFD accuracy loss. Numerically, the MRSE increase from 3-bit quantization is 5.64%, and result of the FFD and the FFG on real face images from LFPW dataset [12] is shown in Fig. 16. Both the FFD result and the FFG results show identical images whether 3-bit quantization is implemented or not. Moreover, tested on a face recognition CNN [6], FFG results utilizing 3-bit quantization did not show accuracy degradation compared to the FFG outputs with non-quantized regression.

Fig. 14(b) shows the power reduction from workload adaptation and clock-gating in 100 MHz frequency. We allocate different number of homogenous PEs to the FFD stage and the FFG stage, and exploit row-by-row parallelism at the FFG stage. The average power consumption is reduced by 37.14% through the proposed workload adaptation and clock-gating scheme.

VI. Conclusion

Despite the improvement of the face recognition accuracy through the advent of deep CNN models, the 3-D face frontalization remains as an essential pre-processing stage in the face recognition pipeline. Moreover, resolution of the input images vary with respect to the target application; therefore, the scalability of the architecture to deal with varying image resolution is an important aspect. Therefore, a low-power scalable 3-D face frontalization processor is proposed for highly accurate mobile face recognition system.

To resolve the unbalanced workload between the stages in the face frontalization algorithm and high external memory access, this paper proposes three key features. For the reduction of EMA, accuracy scalable regression weight quantization based on K-means clustering, and pipelined memory-level zero-skipping regression is proposed. From the proposed two EMA reduction schemes, 99.7% EMA decrease is achieved. For unbalanced workload of the 3-D face frontalization algorithm, workload adaptation through PE allocation with row-by-row parallelism and clock-gating is proposed, from which 37.14% of power reduction is achieved. As a result, 3-D face frontalization processor with 4.73 fps framerate with 0.53 mW power consumption is successfully implemented.

References

[1] *Samsung Galaxy S8*. Accessed: Jun. 14, 2018. [Online]. Available: https://www.samsung.com/global/galaxy/galaxy-s8/

[2] B. Mandal, S.-C. Chia, L. Li, V. Chandrasekhar, C. Tan, and J.-H. Lim, "A wearable face recognition system on Google glass for assisting social interactions," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014.

[3] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang. (2015). "Targeting ultimate accuracy: Face recognition via deep embedding." [Online]. Available: https://arxiv.org/abs/1506.07310

[4] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep. 07-49, Oct. 2007, vol. 1, no. 2.

[5] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1701–1708.

[6] K. Bong, S. Choi, C. Kim, S. Kang, Y. Kim, and H. J. Yoo, "A 0.62 mW ultra-low-power convolutional-neural-network face-recognition processor and a CIS integrated with always-on Haar-like face detector," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 248–249.

[7] *Samsung Exynos 5422*. Accessed: Jun. 14, 2018. [Online]. Available: http://www.samsung.com/semiconductor/minisite/exynos/products/mobileprocessor/exynos-5-octa-5422

[8] E. Zangeneh, M. Rahmati, and Y. Mohsenzadeh. (2017). "Low resolution face recognition using a two-branch deep convolutional neural network architecture." [Online]. Available: https://arxiv.org/abs/1706.06247

[9] C. Herrmann, D. Willersinn, and J. Beyerer, "Low-resolution convolutional neural networks for video face recognition," in *Proc. 13th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, 2016, pp. 221–227.

[10] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 815–823.

[11] T. Hassner, S. Harel, E. Paz, and R. Enbar, "Effective face frontalization in unconstrained images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4295–4304.

[12] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 532–539.

[13] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face alignment at 3000 FPS via regressing local binary features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1685–1692.

[14] M. Horowitz, "Computing's energy problem (and what we can do about it)," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2014, pp. 10–14.

[15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015.

[16] A. Dempster, M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Stat. Soc. B, Stat. Methodol.*, vol. 39, no. 9, pp. 1–38, 1977.

[17] L. Kaufman and P. J. Rousseeuw, *Clustering by Means of Medoids*. Haarlem, The Netherlands: North-Holland, 1987.

[18] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.

[19] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, "Interactive facial feature localization," in *Computer Vision—ECCV*. Berlin, Germany: Springer, 2012, pp. 679–692.

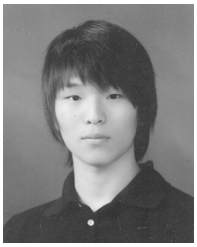
[20] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2930–2940, Dec. 2013.

Sanghoon Kang (S'16) received the B.S. degree from the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2016, where he is currently pursuing the M.S. degree.

His research interests include low-power vision SoC design and deep learning processor design.

Jinmook Lee (S'15) received the B.S. degrees in electrical engineering from Hanyang University, Seoul, South Korea, in 2014, and the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2016, where he is currently pursuing the Ph.D. degree.

His research interests include micro-architecture for low-power SoC and deep learning algorithm for sequence recognition.



Kyeongryeol Bong (S'12) received the B.S. and M.S. degrees from the School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2012 and 2015, respectively, where he is currently pursuing the Ph.D. degree.

His current research interests include low-power vision SoCs especially focused on deep learning processor design and functional CMOS image sensor design.

Changhyeon Kim (S'16) received the B.S. and M.S. degrees from the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree.

His research interests include low power vision SoC design using CMOS image sensors.

Youchang Kim (S'12) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2012, 2014, and 2017, respectively.

He is currently a Post-Doctoral Researcher with the Information and Electronics Research Institute, KAIST. His research interests include energy-efficient multi-core systems especially focused on low-power network-on-chip, near-threshold circuit, machine learning, and artificial intelligence.

Dr. Kim received the Grand Prix from the Ministry of Knowledge Economy at Wearable Computer Contest in 2010, the IEEE Solid-State Circuits Society (SSCS) Predoctoral Achievement Award 2015–2016, and the IEEE International SSCC 2016 Demonstration Session Certificate of Recognition.

Hoi-Jun Yoo (M'95–SM'04–F'08) graduated from the Electronic Department of Seoul National University, Seoul, South Korea, in 1983, and received the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, in 1985 and 1988, respectively.

Since 1998, he has been a faculty member with the Department of Electrical Engineering, KAIST, where he is currently a Full Professor. From 2001 to 2005, he was the Director of the Korean System Integration and IP Authoring Research Center. From 2003 to 2005, he was the full time Advisor to the Minister of Korea Ministry of Information and Communication and National Project Manager for SoC and Computer. In 2007, he founded the System Design Innovation and Application Research Center, KAIST. Since 2010, he has been serving as the General Chair of the Korean Institute of Next Generation Computing. His current interests are computer vision SoC, body area networks, and biomedical devices and circuits. He has co-authored *DRAM Design* (Korea: Hongrung, 1996), *High Performance DRAM* (Korea: Sigma, 1999), *Future Memory: FRAM* (Korea: Sigma, 2000), *Networks on Chips* (Morgan Kaufmann, 2006), *Low-Power NoC for High-Performance SoC Design* (CRC Press, 2008), *Circuits at the Nanoscale* (CRC Press, 2009), *Embedded Memories for Nano-Scale VLSIs* (Springer, 2009), *Mobile 3D Graphics SoC from Algorithm to Chip* (Wiley, 2010), *Bio-Medical CMOS ICs* (Springer, 2011), *Embedded Systems* (Wiley, 2012), and *Ultra-Low-Power Short-Range Radios* (Springer, 2015).

Dr. Yoo received the Electronic Industrial Association of Korea Award for his contribution to DRAM technology in 1994, the Hynix Development Award in 1995, the Korea Semiconductor Industry Association Award in 2002, the Best Research of KAIST Award in 2007, the Scientist/Engineer of this month Award from the Ministry of Education, Science and Technology of Korea in 2010, the Best Scholarship Award of KAIST in 2011, and the Order of Service Merit from the Ministry of Public Administration and Security of Korea in 2011. He was a co-recipient of the ASP-DAC Design Award in 2001, the Outstanding Design Awards of 2005, 2006, 2007, 2010, and 2011, the 2014 A-SSCC, the Student Design Contest Award of 2007, 2008, and 2010, and the 2011 DAC/ISSCC. He has served as a member of the Executive Committee of ISSCC, Symposium on VLSI, and A-SSCC. He also served as the TPC Chair of the A-SSCC in 2008, ISWC in 2010, and ISSCC in 2015, an IEEE Distinguished Lecturer from 2010 to 2011, the Far East Chair of ISSCC from 2011 to 2012, the Technology Direction Sub-Committee Chair of ISSCC in 2013, and TPC Vice Chair of ISSCC in 2014.