Cost-Efficient Super-Resolution Hardware Using Local Binary Pattern Classification and Linear Mapping for Real-Time 4K Conversion

SUNG-HO BAE, (Member, IEEE), JAE-HYEONG BAE, ABDUL MUQEET, MST SIRAZAM MONIRA, AND LOKWON KIM
Department of Computer Science and Engineering, Kyung Hee University, Yongin-si 17104, South Korea

Corresponding author: Lokwon Kim (lwk@khu.ac.kr)

This work was supported in part by the Technology Innovation Program or Industrial Strategic Technology Development Program funded by the Ministry of Trade, Industry & Energy (MOTIE, South Korea) (Memristor Fault-Aware Neuromorphic System for 3D Memristor Array) under Grant 10085646, and in part by the Korean Ministry of Science and ICT (MSIT), through the National Program for Excellence in SW, supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP) under Grant 2017-0-00093.

ABSTRACT We propose a new hardware-friendly super-resolution (SR) algorithm using computationally simple feature extraction and regression methods, i.e., local binary pattern (LBP) and linear mapping, respectively. The proposed method pre-trains dedicated linear mapping kernels for different texture types of low-resolution (LR) image patches where the texture type is classified based on LBP features. On inference operation, a high-resolution (HR) image patch is reconstructed by multiplying an LR image patch with a linear mapping kernel, which is inferred by the LBP feature class of the corresponding LR patch. Since, the LBP is a highly efficient feature extraction operator for local texture classification, our method is extremely fast and power-efficient while showing competitive reconstruction quality to the latest machine learning-based SR techniques. We also present a fully pipe-lined hardware architecture and its implementation for real-time operations of the proposed SR method. The proposed SR algorithm has been implemented on a field-programmable-gate-array (FPGA) platform including Xilinx KCU105 that can process 63 frames-per-second (fps) while converting full-high-definition (FHD) images to 4K ultra-high-definition (UHD) images. Extensive experimental results show that the proposed proposed algorithm and its hardware implementation can achieve high reconstruction performance compared to the latest machine-learning-based SR methods while utilizing minimum hardware resources, thereby having remarkably less computational complexity. Sometimes, the latest deep-learning-based SR approaches offer slightly higher reconstruction quality, but they require significantly larger amount of hardware resources than the proposed method.

INDEX TERMS Local binary pattern, real-time, super-resolution, UHD.

I. INTRODUCTION Recently, televisions (TVs) and smartphones which adopt UHD resolution (3840 × 2160) displays are rapidly becoming the mainstream in the market. However, most of the video contents consumed today are produced with the conventional FHD resolution (1920 × 1080). In order to reproduce these LR images on UHD displays, the LR images must be upsampled to the UHD resolution [18].

Image SR is an up-scaling technique that maps LR images into HR ones while improving visual quality [10]–[12], [28]. In general, interpolation-based up-scaling methods (e.g., bicubic interpolation method) are widely used as an SR method [30]. This is because interpolation-based up-scaling methods are computationally very efficient and hardware-friendly. However, they often induce blurry or jaggy artifacts, thereby degrading the quality of resulted images [33].

Recently, numerous machine learning and deep learning based methods are proposed [30]. However, most of these learning-based approaches require a huge amount of
computations which make it unsuitable for real-time and low-power applications.

To overcome the huge computational complexity of learning-based SR methods, initial-upscaling method is removed and SR output is directly recovered by using LR image [1], [4], [5], [23]. In [1] an edge-orientation (EO) based method is proposed. In contrast to conventional approach, they removed the initial up-scaling step and apply reconstruction on LR image and then, they linearly map the 3 × 3 patches into 2 × 2 patches, which are indexed based on their gradients’ magnitude and directions.

In this paper, we propose a hardware-friendly SR algorithm and hardware implementation which achieves comparable reconstruction quality of the ScSR method [31] but still has similar computational complexity compared to bicubic interpolation method i.e., it is suitable for real-time applications.

Our approach differs in mapping step and generates dedicated linear SR mapping by extracting a local image structure with LBP classification. LBP has widely been used in texture classification tasks [19] but has hardly been used in the SR. To the best of our knowledge, this paper is the first to show that LBP exhibits considerable effectiveness with linear mapping methods. Furthermore, we implemented the proposed SR hardware implementation on an FPGA for practical applications requiring real-time HR image generations. Our implementation can process 63 frames-per-second (fps) in case of FHD to UHD conversion with very few hardware resources.

Our main contributions are as follows:

1) We propose an SR algorithm which generates dedicated linear SR mapping by extracting a local image structure with LBP classification.
2) Also, we develop a hardware-friendly implementation that can process 63 fps in case of FHD to UHD conversion with limited hardware resources.
3) The extensive experiments of our both software and hardware implementations on multiple datasets verify the superiority of the proposed method in terms of visual quality as well as inference time.

The rest of the paper is organized as follows: Section II further describes previous SR approaches and their hardware implementations. Section III explains mathematical details of the proposed LBP based SR method. Section IV presents an in-depth look at the hardware implementation of the proposed SR algorithm for real-time operations. Section V verifies the effectiveness of the proposed method via comprehensive experiments in terms of both the reconstruction quality and hardware implementation efficiency. Finally, Section VI summarizes the paper.

II. RELATED WORKS

Early SR methods considered as traditional signal-processing based approaches which rely on interpolation based algorithms. Specifically, they make use of various interpolation kernels such as bilinear or bicubic kernels to reconstruct a high resolution image from its counterpart LR image. They are easy to implement in hardware because these methods require only a few line of buffers and are low in the computational complexity. Many modified versions of interpolation-based SR methods have been proposed [1], [8], [15] where [15] is the hardware-friendly implementation based on [1].

However, interpolation methods introduce artifacts in the up-scaled images. In order to overcome the artifact problem, Wang et al. [29] proposed an up-scaling method that dynamically integrates a sharpness filter to enhance the visual quality of reconstructed HR images compared to traditional interpolation-methods. It reconstructs HR images while maintaining considerable sharpness, thus perceptually improving visual quality. But, the computational overhead for the dynamic sharpening procedure is larger than the interpolation procedure that makes it two to three times slower than traditional interpolation-methods despite having a marginal improvement of reconstruction quality.

On the other hand, in machine learning based approaches, the missing information is restored by referencing pre-trained kernels or internal/external exemplars of input images instead of interpolating form nearby pixel values. Yang et al. proposed ScSR that jointly trains LR and HR dictionary pairs [31]. For inference, it solves an optimization problem aiming at finding the best set of sparse coefficients from the LR dictionary and performs non-linear mapping for the set of sparse coefficients to the HR dictionary. Finally, HR images are reconstructed with a linear combination of the sparse coefficients and their corresponding HR dictionary atoms. Huang et al. [9] proposed a self-exemplar based SR method (SelfEx) that utilizes repeating structures in an input image. First, it upscales the input LR image to an HR image and searches for the best image patch in the LR image from the HR image patch with respect to minimizing reconstruction errors. In order to find the best self-image patch, SelfEx performs extensive searching with various affine-transformations for a single image patch. Although these approaches show much better reconstruction results, they have excessively high computational complexity due to solving the optimization problems, which makes them impractical to use in real-time applications.

Recently, CNN-based SR methods have shown promising results due to their complex feature representation ability. Dong et al. [3] proposed a CNN-based SR method (SRCNN) that is considered as the milestone work in the SR research field. SRCNN consists of three convolution layers where the convolution filters are learned to minimize the mean squared errors between the target HR and its predicted HR images. But SRCNN does not converge during the training process when the number of convolution layers is greater than 3. Kim et al. [14] found out that deeper convolution layers for SR cause gradient exploding/vanishing problem that does not allow the CNN-based SR methods to converge during the training. To address this problem, they applied the residual learning and gradient clipping techniques which
prevent the model from gradient exploding/vanishing and make the gradients steadily back-propagated during the training process. As a result, Kim et al. proposed a very deep neural network-based super-resolution method (VDSR) that is comprised of 20 convolution layers, showing impressive reconstruction performance. Tai et al. proposed recursive blocks in DRRN [26] and memory block in MemNet [27]. However, these methods need to interpolate the LR inputs to the desired size first, which inevitably causes information loss and also increases computational burden.

A few attempts have taken on generative adversarial network (GAN) [6] based SR. Ledig et al. [17] proposed SRGAN where they have incorporated a perceptual loss [13] and a GAN loss for photo-realistic SR. Sajjadi et al. [25] introduced a GAN based model, called EnhanceNet, by combining an automated texture synthesis and perceptual loss. Although SRGAN and EnhanceNet can alleviate the blurring and oversmoothing artifacts to a certain extent, they produce unpleasing artifacts in the reconstructed images.

Recently, Zhang et al. proposed a residual dense network (RDN) [34] and a residual channel attention network (RCAN) [35]. RDN encapsulates the dense connection within the residual block whereas RCAN employs the channel attentions mechanism within the residual blocks and uses the residual in residual structure to deepen the network architecture. However, the number of parameters increases with the number of layers and thereby increases the computational complexity. As a result, even though CNN-based SR methods have high reconstruction performance, their applicability is limited due to excessive computational costs, memory requirements, hardware resource requirements, and power consumption.

To overcome the huge computational burdens of learning-based SR methods, Choi and Kim [1] proposed an EO (Edge Orientation)-based linear mapping SR method, namely super-interpolation (SI). By employing the EO characterization with SR-optimized multiple-linear-mapping kernels, it conducts SR images with factor of 2 by linearly mapping a 3 × 3 LR patch to a 2 × 2 HR patch. Compared to other learning-based SR hardware, the hardware implementation based on SI method called HSI requires substantially less hardware resources while achieving similar performance to that of ScSR [15]. However, because it uses 625 linear mapping kernels, the size of block memory (BRAM) required for hardware implementation is considerably large. For low-cost mobile applications, such hardware resource requirements can be demanding. Also, the degree of edge orientation should be quantized, which limits the representation power for EO features and requires finding the optimal number of quantization steps.

Several steps have been taken towards the hardware implementations for super-resolution. Huang et al. [8] proposed a hardware architecture which implements the bicubic interpolation with adaptive clamping and sharpening spatial filters. It has shown a promising improvement of the quality compared to bicubic interpolation but is still far behind learning-based methods. Manabe et al. [20] proposed an FPGA implementation and performance evaluation of a CNN-based SR to process moving images in real time. Rather than applying pre-enlargement techniques, they used horizontal and/or vertical flips to network input images to prevent information loss. Later, Manabe et al. [21] incorporated the residue number system (RNS) in their previous method [20] to reduce resource utilization. Zhuolun et al. [7] proposed another FPGA based super-resolution system that crops each frame into blocks, measures their total variation values, and dispatches them accordingly to a neural network or an interpolation module for upscaling. This method offers a balance of resource utilization, the attainable frame rate, and the image quality.

Recent studies on sparse signal representation suggest that the linear relationships among high-resolution signals can be accurately recovered from their low-dimensional projections [18], [19]. Based on that above-mentioned fact and by following [1], we propose a new hardware friendly SR method that can process real-time images with high reconstruction quality. Specifically, we utilize the linear regression from [1], but incorporates LBP instead of EO for characterizing the local image structure. The LBP, which is one of the most powerful operators for texture classifications, was first proposed by Ojala et al. [24]. It works by taking a 3 × 3 gray level image patch and comparing the value of the center pixel with the outer eight pixels. By assigning 1 for pixels with higher value and 0 for the others, we obtain an 8-bit binary vector which represents the LBP feature of the center pixel. In this way, LBP features can be extracted for every pixel in an image excluding the edge. Its strength is revealed in a situation where change of illumination occurs because each feature is quantized by a relative threshold. LBP has shown great achievements in the field of computer vision and pattern recognition. Another great strength of LBP is its mathematical simplicity which makes it suitable for real-time and low-power hardware operation.

The advantages of LBP are brought to SR by integrating LBP with a linear mapping technique. Because LBP is a feature specialized for texture classifications and shows great performance in that domain, it characterizes local image structure very well. Our design also takes advantage of the hardware-friendly nature of LBP and linear mapping techniques. Although it shows great HR reconstruction quality, the hardware implementation of this SR method uses very low hardware resources. Furthermore, in contrast to the SI method, we can control the trade-off between the computational complexity and reconstruction performance by setting a threshold bypass for homogeneous regions. As a result, our proposed method shows better reconstruction performance than the SI method while having a lower computational complexity.

In summary, by taking the advantages of interpolation, machine-learning, feature extraction, and regression techniques, we propose a very efficient SR method that is comparable to the latest machine-learning based SR methods.
III. PROPOSED ALGORITHM

A. SINGLE IMAGE SUPER-RESOLUTION

SR problem can be mathematically represented by [32].

\[
y = (x \otimes k) \downarrow_{ds} + n,
\]

(1)

where \(x\) is desired SR image and \(y\) is input LR image in Eq. 1. And, \(k\) and \(ds\) are the blur kernel and down-sample operator respectively. By following previous work [15], we assume the bicubic interpolation as a down-sample operator. Lastly, \(n\) refers to additive noise, typically gaussian noise is considered. Since we only know the LR image, this problem becomes an ill-posed problem. However, learning-based methods, such as [2]–[4] trained CNNs to achieve this task. We explain our proposed method in next section. Here, we first explain the proposed software-based SR and then the proposed hardware-based SR.

B. SOFTWARE-BASED SR

Our method is mainly derived from the existing SR method in [1], and replaces the EO extraction process of SI with a more efficient LBP operator. To create an efficient LR-to-HR mapping function for real-time up-scaling, the proposed method is divided into two phases: training and inference.

1) TRAINING

The procedure of generating training data is described in Section V. This training operation generates pre-trained kernel values that would be used at later stages, i.e., for inference. For the training, first, we extract the LBP features values from LR patches by creating 256 clusters. These features values are ranging from 0 to 255 each showing different texture class. For each texture class in 256 clusters, we construct a simple mapping function from LR-to-HR that approximates the original HR patches by training a linear mapping kernel matrix for each texture class using the following equation

\[
M_C = X_C Y_C^T \left( Y_C Y_C^T + \lambda I \right)^{-1}
\]

(2)

where \(M_C\) denotes the mapping kernel matrix for each texture class \(C\). The trained kernel matrix \(M_C\) is used for testing or inference stage. \(I\) is an identity matrix, \(Y\) is a concatenate matrix of a set of vectorized LR patches and \(X\) is the corresponding matrix of vectorized HR patches in the texture class \(C\).

In 2 the \(\lambda\) constant acts as regularizer. We set \(\lambda\) to 1. We experimented with the different \(\lambda\) values ranging from 1 to 10 and observed that the performance remains same, which shows that the performance of our SR method is not sensitive to \(\lambda\).

2) INFERENC

The inference step is designed to be very computationally simple and can be easily implemented in low-cost hardware for real-time up-scaling. For the faster inference, we use \(3 \times 3\) to \(2 \times 2\) mapping to map from LR patches to HR patches as shown in Figure 1.

There can exist \(N = 2 \times M = 2\) possible LR patches in an image sized of \(N \times M\). Therefore, for the given LR image, we can perform \(2 \times SR\) by inferencing one HR patch for each LR patch. Thus, the resulting HR image has a size of \(2N - 4 \times 2M - 4\). Figure 1 shows the \(3 \times 3\) to \(2 \times 2\) SR operation. First, the pixel window scans the LR image top-left to top-right. Then after finishing the first line, the window scans the next line from left to right. This operation is continued until it meets the bottom-right edge of the image, completing the up-scaling operation of an image. The boundary pixels are up-scaled using the nearest-neighbor technique for our hardware implementation.

For the kernel multiplication operation, the \(3 \times 3\) LR patch is flattened to a \(1 \times 9\) vector. This LR vector is multiplied with obtained pre-trained kernel \(M_C\) that is the linear mapping kernel of the LBP class extracted from the LR patch. This yields the reconstructed vectorized HR patch \(\hat{y}\). This procedure is presented in Figure 2. The equation for kernel multiplications in the inference operation can be shown as

\[
x_{i,c} = M_C y_{i,c},
\]

(3)

where \(x_{i,c}\) denotes the HR reconstruction from \(y_{i,c}\) which represents the \(i\)-th patch of the LR input of LBP class \(C\). By performing inference operations for all possible LR patches from an LR image, we obtain a fully reconstructed HR image of the input LR image.

However, mapping all LR patches to HR with LBP-based linear mapping can induce unnecessarily sharp or high-frequency results. Thus, we propose a bypass operation for...
LR patches having simply homogeneous textures. This operation significantly helps to reduce the computational costs of the proposed SR method while keeping good reconstruction performance. This operation is fully described in Section III-C.

**C. THE LBP CLASSIFICATION AND BYPASS OPERATION**

The LBP feature is extracted for the Y-channel LR patches to classify their local texture characteristics (e.g., edge direction). Figure 3 shows the operation of finding an LBP class number from a given LR patch. Firstly, all the surrounding pixel values of $3 \times 3$ LR patch ($i$) are subtracted by the center pixel value $i_C$. Then, according to the sign of each subtracted result, 0 if negative or 1 if positive is assigned to the $i$-th place of an 8-bit binary vector. This resulting binary value represents the LBP class number of the given patch.

To maximize the efficiency of our SR method and control the trade-off between the computation cost and reconstruction performance, we proposed the bypass operation. We also found that, for homogeneous texture patches, up-scaling operation using bilinear interpolation can show better reconstruction results compared to the LBP-based linear mapping method. To determine whether a patch has homogeneity in texture or not, the difference value is calculated for each patch. The threshold value is predefined in the training phase. It is worth noting that the trade-off between computation costs and reconstruction performance can be adjusted using the threshold in the bypass operation. We perform comprehensive experiments to see the effectiveness of the threshold value as shown in Figure 8 in Section V. By calculating the average PSNR value of test set inference results for each different threshold values, we can select the value where the maximum quality was achieved. Detailed information on the test methods and results are explained in Section V.

**IV. PROPOSED HARDWARE**

The proposed hardware architecture is designed to achieve real-time conversion at 60 fps in 1080p (1,920 × 1,080) to 4K (3,840 × 2,160) SR operation. Input and output image streams are composed of each Y-, Cb- and Cr-channels. The up-scaling operation for Y-component goes through the pipeline dedicated for LBP based SR method. Cb- and Cr-components are fed to the pipeline for bicubic interpolation. Because the Y component is far more perceptually sensitive to human eyes, the use of a less expensive up-scaling method which is the bicubic interpolation is appropriate for Cb- and Cr-components [1]. Because bicubic interpolation only requires multiplications on a small set of constants (A multiplication with a variable operand and a constant one is implemented with a number of adders in hardware implementations), actual multipliers are not needed for Cb and Cr-channel pipeline blocks, which significantly reduces hardware utilization. Our design is capable of handling a continuous stream of video data even without any blanking time in between lines or frames.

The hardware pipeline for the luma channel SR consists of a frame reader, an LBP classifier, a weight fetcher, a matrix multiplier, and a frame writer as shown in Figure 4. In a single clock cycle, one 8-bit LR luma pixel is fed and four 8-bit HR luma pixels are produced. The pipeline is operated with a 3-state finite state machine (FSM). When one of the first line or last two lines of LR image are fed, the state in the FSM becomes ‘boundary’. When any of the line except the top and bottom boundaries is inserted, the state becomes ‘middle’. On horizontal vertical blanking, it becomes ‘idle’ state.

The frame reader module is shown in Figure 5. This module includes two line-buffers and a $3 \times 3$ pixel window which consists of nine 8-bit registers. Each line buffer is implemented with dual port BRAMs. The input buffer in which the current LR pixel is input gets switched on every line incrementation. When the FSM is in the ‘boundary’ state, the two input line buffers get directly filled with input LR pixels. In the ‘middle’ state, every part of the input LR image is scanned by a $3 \times 3$ pixel window. The pixel window works as follows. On every clock, data in all registers are moved to the next column. An LR pixel data is written to the input of third row. The pixel data of the same horizontal location from the last two lines are fetched from the two input buffers and are fed to the input of top two rows of the pixel window. The line buffer containing the older line data is updated with

\[
\text{Diff} = \sum_{n=1}^{8} D_n, \quad (4)
\]

where $\text{Diff}$ is the difference value and $D_n$ is the absolute value of the element-wise subtracted difference as shown in Figure 3.
new pixel data from the output of the third row of the pixel window. Then the $3 \times 3$ patch is sent to the LBP classifier and matrix multiplier for further calculations. Because the LBP feature of the image boundary cannot be extracted, boundary pixels are up-scaled with the nearest-neighbor algorithm. Therefore, the boundary pixels read from the frame reader module is forwarded directly to the frame writer module.

The LBP classifier module is shown in Figure 6. This module generates an LBP feature class index based on the $3 \times 3$ patch from the pixel window. An adder is placed for every boundary pixel in the $3 \times 3$ patch to obtain the difference value between itself and the center pixel. It also determines the value of SR bypass flag by comparing the sum of eight absolute difference values with a given threshold. If the sum is smaller than the threshold, SR bypass flag is set to 1. When the flag is up, the matrix multiplier module will perform linear interpolation for the corresponding patch instead of multiplying a linear-mapping kernel.

The weight fetcher module consists of a look-up table (LUT) which contains pre-trained linear-mapping kernels. There are 256 kernel entries in the LUT and each LBP class is mapped to a kernel. Thus, the LBP feature class is an 8-bit binary vector. In addition, each kernel is a $9 \times 4$ matrix consisting of 36 elements (the bit width of each element is 11 bits.). To implement the LUT, a straightforward manner can design 396 bit width with 256 entries in the FPGA BRAM which require 6 BRAMs in the FPGA. To save the BRAM HW resource in the FPGA, we split each kernel value (396 bits) to two 198-bit values, and designed the LUT to be 198 bit width with 512 entries in the FPGA BRAM. Then, we inserted the doubled high clock frequency (compared to the rest of pipeline) to the BRAM containing the LUT so that the pipeline design can read 396 bit kernel value in a single clock cycle. In doing so, we could use half of the BRAMs (from 6 to 3 BRAMs) for the linear-mapping kernels.

The matrix multiplier module multiplies $3 \times 3$ pixel patch with the corresponding kernel. It also performs linear interpolation for sub-threshold patches. If the bypass flag is low, the result of the weight matrix multiplication is produced while if the bypass flag is high, the linearly interpolated result is produced. The multipliers used in this module are constructed of 36 DSP blocks in the FPGA. These DSP blocks are pipelined in 3 stages to increase performance and reduce power consumption.

The HR result from the matrix multiplier module is then sent to the frame writer module where the results get written to a pair of two line buffers. These buffers store two lines of output video data and are interchanged every line increment. While one pair of output buffers are being written, the others are read for output.

A. CHROMINANCE CHANNEL PIPELINE

The chrominance channel up-scaling pipeline consists of a frame reader, a matrix multiplier, and a frame writer. The frame reader module consists of 2 line buffers and a $4 \times 4$ pixel window. The operation of this module is very similar to the frame reader of the luma channel pipeline. $4 \times 4$ patch from the pixel window is sent to the matrix multiplier module where the bicubic interpolation kernel is applied. Since the kernel is a $7 \times 7$ matrix consisting of 7 constants, it does not need actual multipliers or DSP blocks to implement in hardware. The frame writer module receives HR results and writes them to $2 \times 2$ output line-buffers.
V. EXPERIMENTAL RESULTS

To train the proposed model, we use the same training data that are used in [31], i.e., T91 which contains 91 images. We obtained corresponding LR images by down-scaling using the bicubic interpolation with a scale factor of 0.5. Direct mapping method was used to generate the LUT of 256 linear-mapping kernels in floating-point numbers. MATLAB was used for training and testing of the floating-point operation. For hardware implementation, each element of the kernels was quantized to an 11-bit vector. We created a testing program which operates with only fixed-point arithmetic in C to measure the PSNR and SSIM [36] values of the hardware implementation.

To evaluate the proposed method, we perform experiments on five widely used benchmark datasets i.e., Set5, Set14, Kodak, Urban100 and BSDS100. Set5, Set14 and Kodak are considered as smaller datasets since they contain 5, 14 and 24 images, respectively. Both of the Urban100 and BSDS100 contain 100 test images. We use PSNR and SSIM values to compare the reconstruction quality of the proposed SR method (software and hardware) with other methods in comparison. The PSNR and SSIM values of the proposed algorithm were computed using the down-scaled images and the HR images generated with the proposed SR algorithm using floating-point arithmetic.

A. PERFORMANCE COMPARISON

Table 1 presents the performance comparison of the methods in comparison in terms of PSNR and SSIM. We can see that the hardware implementation of the proposed method achieves highest PSNR for Set5 and Kodak datasets and achieves competitive results for other three datasets i.e., Set14, Urban100 and BSDS100. On the other hand, the proposed method achieves highest scores in terms of SSIM for Set5, Kodak and Urban100 datasets while achieves competitive performance for other two datasets i.e., Set14 and BSDS100. As a a result, the reconstruction performance of
TABLE 1. Comparison of reconstruction quality of different Super Resolution methods in terms of PSNR and SSIM. The best performances are highlighted with bold face text. Here HW and SW are referred to as the hardware and software based implementation of the corresponding methods.

| Dataset/Method | Bicubic | ScSR* [31] | HSI (HW) [15] | HSI (SW) [15] | Ours (HW) | Ours (SW) |
|----------------|---------|------------|---------------|---------------|-----------|-----------|
| Set5           | 33.68   | **35.78**  | 34.78         | 34.86         | 34.96     | 34.71     |
| Set14          | 30.33   | **31.64**  | 31.63         | 31.26         | 31.35     | 31.22     |
| Kodak          | 31.42   | 32.21      | 31.24         | 31.27         | 32.36     | 32.24     |
| Urban100       | 26.88   | 28.26      | N/A           | 27.99         | 28.02     | 27.96     |
| BSDS100        | 29.57   | 30.77      | 30.48         | 30.34         | 30.39     | 30.30     |

TABLE 2. The results are evaluated on Set14(×2). We tested on Set14 dataset and report the average time.

| Methods | Bicubic | ScSR | HSI (HW) | HSI (SW) | Ours (HW) | Ours (SW) |
|---------|---------|------|----------|----------|-----------|-----------|
| Run Time (fps) | 3.00 | 0.05 | 0.08 | 0.65 | 3.86 |

TABLE 3. Comparisons of NIQE [22] scores. A smaller score shows a better perceptual quality. Here SW is referred to as the software based implementation of the corresponding methods.

| Methods | Set5 | Set14 | Kodak | Urban100 | BSDS100 |
|---------|------|-------|-------|----------|---------|
| Bicubic | 6.04 | 5.50  | 5.40  | 6.95     | 5.76    |
| ScSR [31] | 5.25 | 5.09  | 4.67  | 6.49     | 4.14    |
| HSI (SW) [15] | 5.39 | 6.40  | 5.98  | 7.64     | 6.97    |
| Ours (SW) | **4.89** | 5.39  | 4.99  | **6.28** | 5.48    |

TABLE 4. Hardware utilization and performance.

| Channel | Luma | Chrominance |
|---------|------|-------------|
| Hardware Utilization | LUT : 1,151 | LUT : 765 |
| Registers : 1,290 | Registers : 284 |
| DSP : 36 | DSP : 0 |
| BRAM : 35KiB | BRAM : 15.75KiB |
| Max. Freq. (MHz) | 154 | 176 |

the proposed method is competitive to the other state-of-the-art methods.

Furthermore, we compare the computational complexity of the methods in comparison and present the results in Table 2. Here we measure the run time of the methods in terms of frame per seconds (fps). As shown in Table 2, the maximum number of frames can be processed by bicubic interpolation method. However the reconstruction quality of bicubic interpolation is poor compared to the other methods. Except the bicubic method, both the hardware and software implementation of the proposed method achieves the highest fps i.e., they can process highest number of frames compared to all other methods in comparison. Specifically, the hardware implementation can process 63 frames per second and the software implementation can process 3.86 frames per second when FHD images are converted to UHD. Also, We found out that the software implementation of our proposed method takes only 1.2 milliseconds (ms) to compute the LBP class as compared to 2.8 ms taken by HSI [15].

Moreover, Figure 7 presents the qualitative results of the methods in comparison. Here, we compare the software-based results. It can be observed that the visual quality of the SR images generated by the proposed method are satisfactory. While bicubic results produce blurry images, our method produces images with similar quality as HSI [15] even with the lower computational complexity. As a result, considering both the reconstruction quality and the inference time, the proposed method can be the optimum choice for real-time super resolution task.

Additionally, we compare the perceptual quality by using the naturalness image quality evaluator (NIQE) [22] and present the experimental results in Table 3. It can be seen that the proposed method consistently outperforms HSI [15] on various datasets. Our method achieves the highest quality for Set5 and Urban100 datasets, while achieves competitive performance in other datasets.

B. FPGA IMPLEMENTATION RESULTS

The proposed design has been integrated into the Xilinx KCU105 FPGA evaluation board. Table 4 shows the hardware resource utilization for the luma and chrominance channel pipelines. The 6 line buffers and kernel LUT of the luma channel pipeline utilized 35KiB of BRAM. 36 DSP blocks were used for multiplying 3 × 3 pixel patch with the corresponding kernel. 40 LUTRAMs were used for 3 × 3 pixel patch delay buffers. For the chrominance channel pipeline, no DSP blocks were used because multiplication with 7 constants was only required for the bicubic interpolation.

The maximum clock was found by Vivado implementation timing analysis results. We programmed a script that finds the maximum clock by modifying timing constraints and running multiple implementation with different timing parameters in parallel. We successfully verified the real-time operation of the hardware architecture implemented in the FPGA platform. The HR images generated from software simulation was put into the BRAMs in the FPGA. By embedding a hardware which compares the output results and the contents
stored in the BRAMs, we could verify that the hardware works exactly same as the software simulation.

Table 5 shows the comparison of hardware resource utilizations among CNN-based, HSI, and the proposed LBP based SR hardware. It shows that the proposed method utilized significantly lower hardware resources than the HSI and CNN-based Hardware. The HSI based hardware implementation requires 5 percent more registers, 36 percent more BRAMs, 200 percent more DSP units and 24.5 percent more LUTs than the proposed hardware implementation while the CNN-based one includes 53 times more registers, 5.8 times more BRAMs, 53 times more DSP units and 41 times more LUTs than the proposed one. The factors that the proposed hardware implementation significantly saves hardware resources are as follows. The smaller kernel size (in comparison to HSI) and double-clock operation of the weight fetcher module contributed to small BRAM usage. Also since we used YUV420 color space as input and output instead of RGB, size of line buffers was reduced resulting in low BRAM usage. In addition, because we did not apply the LBP SR method to the chrominance channel, there was a great reduction in the DSP usage.

VI. CONCLUSION

We proposed a novel hardware-friendly SR algorithm that integrates the LBP-based linear mapping method. Due to the high efficiency and hardware-friendliness, our method has shown promising performance compared to the latest learning-based SR models while having very low computational complexity. We also developed a dedicated hardware implementation for the LBP-based SR algorithm and verified it on a Xilinx KCU105 FPGA board. The implemented hardware architecture is capable of processing 63 frames per second i.e., a real-time processing when converting FHD to 4K UHD SR operation. Because of the hardware friendly characteristics of the proposed SR algorithm, the utilization of hardware resources is remarkably low. Considering the high reconstruction performance and significantly low inference time, the proposed method, both the software based and hardware based implementations, can be an optimum choice for low-power and low-cost consumer display devices such as smart phones, tablet PCs, televisions, and monitors.

REFERENCES

[1] J.-S. Choi and M. Kim, “Super-interpolation with edge-orientation-based mapping kernels for low complex 2× upscaling,” IEEE Trans. Image Process., vol. 25, no. 1, pp. 469–483, Jan. 2015.
[2] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 2, pp. 295–307, Feb. 2016.
[3] C. Dong, C. C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, Sep. 2014, pp. 184–199.
[4] C. Dong, C. C. Loy, and X. Tang, “Accelerating the super-resolution convolutional neural network,” in Proc. Eur. Conf. Comput. Vis. (ECCV) (Lecture Notes in Computer Science), vol. 9906, Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 391–407.

[5] J. Du, L. Wang, A. Gholidpour, Z. He, and Y. Jia, “Accelerated super-resolution MR image reconstruction via 3D densely connected deep convolutional neural network,” in Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM), Dec. 2018, pp. 349–355.

[6] J. Ian Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Proc. 27th Int. Conf. Neural Inf. Process. Syst., Cambridge, MA, USA, vol. 2, 2014, pp. 2672–2680.

[7] Z. He, H. Huang, M. Jiang, Y. Bai, and G. Luo, “FPGA-based real-time super-resolution system for ultra high definition videos,” in Proc. IEEE 26th Annu. Int. Symp. Field-Program Custom Comput. Mach., Apr. 2018, pp. 181–188.

[8] C.-C. Huang, P.-Y. Chen, and C.-H. Ma, “A novel interpolation chip for real-time multimedia applications,” IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 10, pp. 1512–1525, Oct. 2012.

[9] J. Huang, A. Singh, and N. Ahuja, “Single image super-resolution from transfer learning,” in Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR), Dec. 2015, pp. 5197–5206.

[10] K. Jiang, W. Pang, Y. Gao, M. Gu, and J. Jiang, “ATMEN: Adaptive-Threshold-Based multi-model fusion network for compressed face hallucination,” IEEE Trans. Multimedia, vol. 22, no. 10, pp. 2734–2747, Oct. 2020.

[11] K. Jiang, Z. Wang, P. Yi, J. Jiang, J. Xiao, and Y. Yao, “Deep distillation recursive network for remote sensing imagery super-resolution,” Remote Sens., vol. 10, no. 11, pp. 1700, Oct. 2018.

[12] K. Jiang, Z. Wang, P. Yi, G. Wang, T. Lu, and J. Jiang, “Edge-enhanced GAN for remote sensing image super-resolution,” IEEE Trans. Geosci. Remote Sens., vol. 57, no. 8, pp. 5799–5812, Aug. 2019.

[13] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 694–711.

[14] J. Kim, J. K. Lee, and K. M. Lee, “Accurate image super-resolution using very deep convolutional networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 1646–1654.

[15] Y. Kim, J.-S. Choi, and M. Kim, “2X super-resolution hardware using Edge-Orientation-Based linear mapping for real-time 4K UHD 60 fps video applications,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 65, no. 9, pp. 1274–1278, Sep. 2018.

[16] Y. Kim, J.-S. Choi, and M. Kim, “A real-time convolutional neural network for super-resolution on FPGA with applications to 4K UHD 60 fps video services,” IEEE Trans. Circuits Syst. Video Technol., vol. 29, no. 8, pp. 2521–2534, Aug. 2019.

[17] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, “Photo-realistic single image super-resolution using a generative adversarial network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 105–114.

[18] J. Lee and I.-C. Park, “High-performance low-area video up-scaling architecture for 4-K UHD video,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 64, no. 4, pp. 437–441, Apr. 2017.

[19] L. Liu, S. Lao, P. W. Fieguth, Y. Guo, X. Wang, and M. Pietikainen, “Median robust extended local binary pattern for texture classification,” IEEE Trans. Image Process., vol. 25, no. 6, pp. 1368–1381, Mar. 2016.

[20] T. Manabe, Y. Shibata, and K. Oguri, “FPGA implementation of a real-time super-resolution system using a convolutional neural network,” in Proc. Int. Conf. Field-Program. Technol. (FPT), Dec. 2016, pp. 249–252.

[21] T. Manabe, Y. Shibata, and K. Oguri, “FPGA implementation of a real-time super-resolution system with a CNN based on a residue number system,” in Proc. Int. Conf. Field Program. Technol. (ICFP'T), Dec. 2017, pp. 299–300.

[22] A. Mittal, R. Soundararajan, and A. C. Bovik, “Making a completely blind—image quality analyzer,” IEEE Signal Process. Lett., vol. 20, no. 3, pp. 209–212, Aug. 2013.

[23] A. Muquet, M. T. B. Iqbal, and S.-H. Bae, “HRAN: Hybrid residual attention network for single image super-resolution,” IEEE Access, vol. 7, pp. 137020–137029, 2019.

[24] T. Ojala, M. Pietikainen, and D. Harwood, “Performance evaluation of texture measures with classification based on kullback discrimination of distributions,” in Proc. 12th Int. Conf. Pattern Recognit., 1994, pp. 582–585.
ABDUL MUQEET received the B.S. degree from the University of Karachi, Karachi, Pakistan, in 2014. He is currently pursuing the M.S. degree with Kyung Hee University, South Korea. His research interest includes efficient deep learning models for image restoration.

MST SIRAZAM MONIRA received the B.S. degree from the Department of Information and Communication Engineering, Islamic University, Kushtia, Bangladesh, in 2015. She is currently pursuing the M.S. degree with the Department of Computer Science and Engineering, Kyung-Hee University, Suwon, South Korea. Her research interests include efficient deep learning and advanced data augmentation.

LOKWON KIM received the Ph.D. degree from the Electrical Engineering Department, UCLA, where he has conducted research on neuromorphic computing for deep learning, system-on-chip architecture, automated optimization methodology for power, area, and performance of VLSI designs, and reconfigurable computing. Prior to joining the department, he has had over a decade full-time research and industry experience at world leading companies, such as Apple, Cisco systems, IBM T.J. Watson Research Center, Broadcom, and Korea Electronics Technology Institute (KETI), where he has played a key role in design and verification of various computer hardware systems widely used in the world. From 2011 to 2014, he has worked with Cisco Systems where he played a key role in modeling, design, and verification of network router core chipsets. From 2014 to 2017, he was with Apple at Cupertino, CA, USA, where he led design and verification of application processors used in iPhone, iPad, Apple Watch, and so on. He visited a Research Team at the IBM T.J Watson Research Center, in 2010, where was conducting a research project (SYNAPSE) on one of the most successful modern neuromorphic processors (TrueNorth) and has led a research project on a neuromorphic processor for deep belief networks and restricted Boltzmann machine artificial neural network which are popular deep learning models. He is currently an Assistant Professor with the Department of Computer Science, Kyung Hee University. He authored or coauthored ten journal articles, nine conference papers, and nine patents. His research led to successful outcomes of the world fastest processors for the models. He was a recipient of the Cisco Achievement Program Award in the recognition of outstanding employee efforts and achievements at Cisco Systems, in 2011.