Received: 21 February 2025 | Revised: 10 March 2025 | Accepted: 11 March 2025

DOI: 10.1002/jsid.2075

WILEY

SPECIAL SECTION PAPER

BDLUT: Blind image denoising with hardware-optimized look-up tables

Boyu Li, SID Student Member[1] | **Zhilin Ai, SID Student Member**[1] | **Baizhou Jiang**[1] | **Binxiao Huang**[1] | **Jason Chun Lok Li**[1] | **Jie Liu**[2] | **Zhengyuan Tu**[2] | **Guoyu Wang**[2] | **Daihai Yu**[2] | **Ngai Wong**[1]

[1]Department of Electrical and Electronic Engineering, The University of Hong Kong, Pok Fu Lam, Hong Kong

[2]TCL Corporate Research (HK) Co., Ltd., Pak Shek Kok, Hong Kong

Correspondence
Ngai Wong, Department of Electrical and Electronic Engineering, The University of Hong Kong, Pok Fu Lam 99077, Hong Kong.
Email: nwong@eee.hku.hk

Funding information
This research was supported in part by the HKU-TCL Joint Research Centre for Artificial Intelligence and in part by the General Research Fund (GRF) project 17203224 and the Theme-based Research Scheme (TRS) project T45-701/22-R of the Research Grants Council (RGC), Hong Kong SAR.

Abstract

Denoising sensor-captured images on edge display devices remains challenging due to deep neural networks' (DNNs) high computational overhead and synthetic noise training limitations. This work proposes BDLUT(-D), a novel blind denoising method combining optimized lookup tables (LUTs) with hardware-centric design. While BDLUT describes the LUT-based network architecture, BDLUT-D represents BDLUT trained with a specialized noise degradation model. Designed for edge deployment, BDLUT(-D) eliminates neural processing units (NPUs) and functions as a standalone ASIC IP solution. Experimental results demonstrate BDLUT-D achieves up to 2.42 dB improvement over state-of-the-art LUT methods on mixed-noise-intensity benchmarks, requiring only 66 KB storage. FPGA implementation shows over 10× reduction in logic resources, 75% less storage compared to DNN accelerators, while achieving 57% faster processing than traditional bilateral filtering methods. These optimizations enable practical integration into edge scenarios like low-cost webcam enhancement and real-time 4 K-to-4 K denoising without compromising resolution or latency. By enhancing silicon efficiency and removing external accelerator dependencies, BDLUT(-D) establishes a new standard for practical edge imaging denoising. Implementation is available at https://github.com/HKU-LiBoyu/BDLUT.

KEYWORDS

algorithm-hardware co-design, blind denoising, lookup table

1 | INTRODUCTION

Image denoising is a fundamental task in computer vision, aiming to recover a clean image from noisy inputs. Traditional methods, such as the bilateral filter (BF),[1-4] the block-matching and 3D (CBM3D) filter,[5] and multi-channel weighted nuclear norm minimization (MC-WNNM),[6] struggle with complex or non-stationary

Boyu Li and Zhilin Ai contributed equally to this manuscript.

[Correction added on 9 May 2025, after first online publication: the contribution statement for the first 2 authors has been added in this version.]

noise patterns. Recent advances in deep neural networks (DNNs)[7,8] have significantly improved denoising performance by learning complex noise distributions. While DNNs often outperform classical methods, they come with challenges such as high computational costs, large dataset requirements, and the risk of overfitting when faced with unseen noise types.

A key limitation of conventional denoising models lies in their reliance on idealized noise assumptions, particularly additive white Gaussian noise (AWGN). Real-world image noise, however, often exhibits complex, spatially dependent, and non-Gaussian characteristics that deviate significantly from these simplified models. While blind super-resolution (SR) techniques[9]—and by extension blind denoising—address unknown noise distributions through data-driven approaches, existing solutions predominantly depend on computationally intensive DNNs. Recent advances in lookup tables (LUTs),[10-13] particularly Hundred-Kilobyte Lookup Tables (HKLUT),[14] offer a promising alternative by precomputing noise-to-clean mappings to drastically reduce inference-time computational overhead. Though LUT-based methods demonstrate potential for balancing performance with hardware efficiency in tasks like SR, their adaptation to blind denoising remains unexplored, presenting an opportunity to bridge the gap between real-world noise complexity and resource-efficient processing.

In this work, we develop BDLUT(-D), a novel approach that integrates blind denoising with LUT-based processing. Our method efficiently handles noise uncertainty while leveraging LUTs for real-time embedded applications. Through extensive experimentation, as shown in Figure 1, we demonstrate that BDLUT(-D) achieves competitive denoising performance with minimal computational overhead, making it a promising solution for practical image denoising task. The key contributions of this work are as follows:

- The first integration of blind denoising with LUT-based processing, validated through extensive experiments on adequate benchmarks and real-world datasets.
- Enhanced kernel design based on HKLUT for improved denoising performance and hardware efficiency, achieving state-of-the-art (SOTA) performance with only 66 KB storage.
- Efficient FPGA implementation and verification, achieving significant resource savings and speed improvements compared to conventional DNN and BF accelerators.

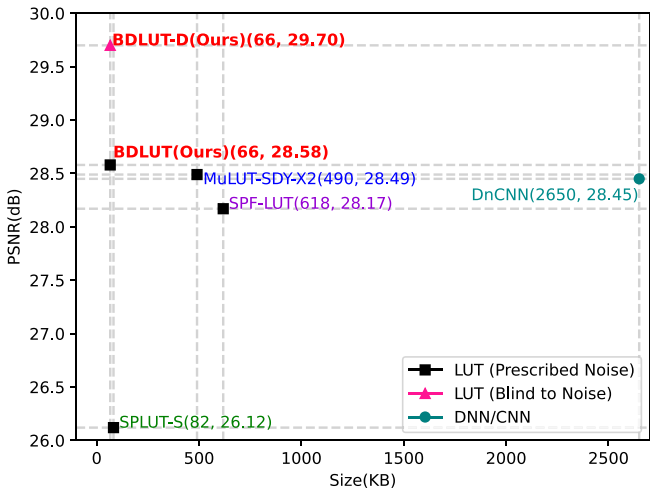


FIGURE 1 Comparison of PSNR (Test set: images from Kodak24 are evenly divided into three groups, each corrupted by AWGN with noise levels of $\sigma = 15, \sigma = 25$, and $\sigma = 50$. Train set: DIV2 K with $\sigma = 15$ AWGN for non-blind LUT) and model sizes between BDLUT(-D) and other schemes.

2 | RELATED WORK

2.1 | Classical approaches

Classical denoising methods, such as the BF,[1-4] BM3D,[5] and MC-WNNM,[6] have long been utilized to improve image quality. The BF method preserves edges while reducing noise by applying a combination of domain and range filtering. Hardware implementations of BF, using techniques like LUTs or specialized filters, have been proposed to accelerate the denoising process. Figure 2A illustrates the traditional BF denoising method: The image is first corrupted with Gaussian noise of a certain intensity and then denoised using the BF algorithm.[2] Additionally, optimization techniques such as approximate computing and kernel isotropic transformation have been explored to improve the hardware implementation of BF.[1,3,4] The speed of BF hardware is inherently constrained by the intensive multiplication operations it requires. Moreover, achieving high denoising accuracy with the BF algorithm demands substantial computational resources, which significantly limits its scalability. BM3D improves performance by leveraging block similarity between image regions, but its high computational complexity poses challenges for efficient hardware implementation.

2.2 | Deep learning approaches

Figure 2B depicts a conventional deep learning-based denoising framework, where a DNN is trained to remove Gaussian noise of predetermined intensity from corrupted images. Researchers have also explored deploying such DNNs on FPGAs for real-time denoising applications. Deep learning models, such as DnCNN[8] and

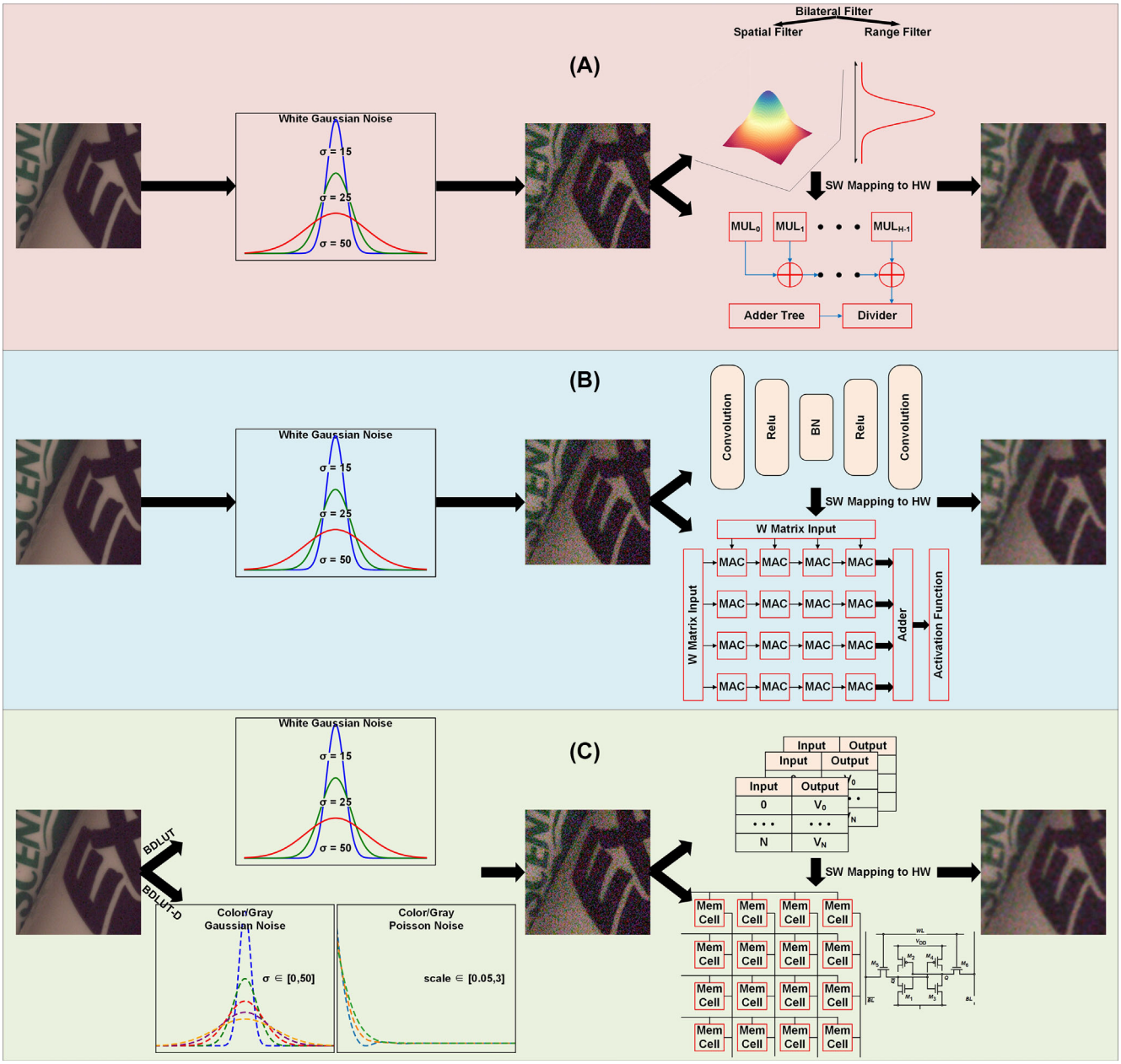


FIGURE 2 Denoising schemes: (A) classical BF; (B) DNN/CNN; (C) LUT-based network: upper branch without noise degradation model (BDLUT), versus lower branch with noise degradation model (BDLUT-D) where the noise is added in two successive stages, each stage can be either Gaussian or Poisson noise chosen by a prescribed probability.

SwinIR,[7] have achieved state-of-the-art (SOTA) denoising performance by learning intricate noise patterns from extensive datasets. While DnCNN employs residual learning to isolate noise efficiently, SwinIR leverages a hierarchical attention mechanism for enhanced image restoration. Despite their effectiveness, these models demand substantial computational resources, including memory and processing power, rendering them impractical for resource-constrained embedded systems. Efforts to mitigate these challenges include the CNN denoising accelerator,[15] which simplifies DnCNN into a lightweight variant (Light-DnCNN) and optimizes hardware integration for reduced data transfer overhead. However, experimental evaluations reveal that even this streamlined implementation consumes significant FPGA logic resources, leads to high power consumption, and introduces substantial processing delays, limiting its viability for low-power edge applications.

2.3 | LUT-based approaches

For image processing tasks, LUT-based methods[10-14] offer a significant advantage over traditional and DNN-

based methods by maintaining high performance while dramatically reducing computation and improving computational efficiency. These methods minimize the need for complex calculations by pre-computing values and storing them in memory for rapid retrieval. For instance, MuLUT[11] uses multiple LUTs in parallel to enhance the receptive field and improve denoising quality. LUT-based approaches are particularly appealing for hardware implementation due to their low computational cost, simplicity, and fast processing speed. However, the application of LUTs specifically for denoising and their hardware implementation has not been widely explored. Figure 2C illustrates our proposed LUT-based denoising method. The upper branch shows the traditional approach of adding Gaussian noise, while the lower branch integrates degradation models (random noise) and applies LUT-based algorithms for denoising, in both software and hardware implementations. The hardware-accelerated LUT version achieves high-quality denoising with significantly reduced computational overhead, highlighting the advantages of LUT-based processing.

2.4 | Degradation models

Noise degradation (or simply degradation) models[16-18] are crucial for training denoising models and simulating real-world noise characteristics. These models enhance the ability of denoising methods to generalize across a wide range of noise types and degradation scenarios. Our approach incorporates a degradation model, allowing our LUT-based method to effectively handle more complex and variable noise, thereby increasing its adaptability for practical applications.

BDLUT(-D) is a hardware-optimized LUT-based denoising solution that integrates blind denoising with efficient hardware processing. Unlike classical and DNN-based methods, which assume specific noise models and require high computational resources, BDLUT(-D) can handle various types of noise without prior knowledge, making it more flexible and efficient. By using LUTs, we significantly reduce computational overhead, enabling real-time performance on embedded systems. Our method demonstrates superior efficiency and speed, particularly in resource-constrained environments, offering a promising solution for practical image denoising tasks.

3 | DEVELOPMENT OF BDLUT

3.1 | Optimized kernel design

LUT has been widely used in the field of image SR, which has good learning abilities and is more hardware-friendly. In order to achieve better denoising results while being more suitable for hardware computation, we have made some improvements based on the HKLUT model.[14]

The overall BDLUT model architecture and corresponding CNN structure is visible in Figure 3: The 8-bit input noisy image is split into two branches (i.e., 4 most significant bits [MSBs] and 4 least significant bits [LSBs] as in SPLUT[19]) for processing. At inference, the CNN network with a first convolutional layer kernel size of K is converted to its corresponding LUT block. After rotation ensemble and addition, the output image of previous module is fed into the subsequent module for the same operation to achieve further noise reduction, creating a cascade structure. The final output ($\hat{y}_i$) for a single BDLUT branch is calculated using the following formula:

$$\hat{y}_i = \frac{1}{N}\frac{1}{M_k}\sum_{k=0}^{N}\sum_{j=0}^{M_k} R_j^{-1}(LUT_k(R_j(x_i))) \quad (1)$$

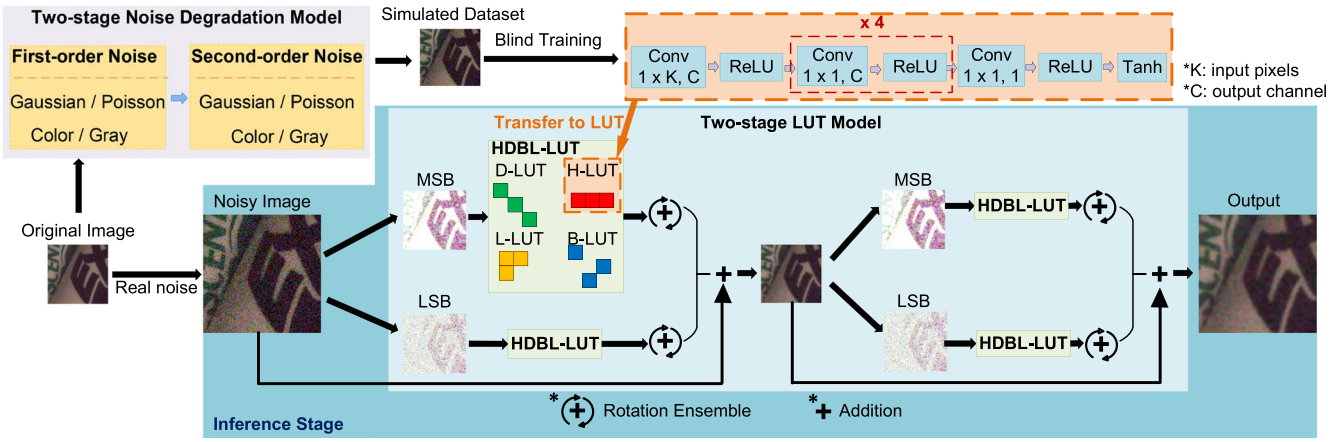


FIGURE 3 BDLUT architecture featuring dual-branch LUT structure with rotation ensemble. BDLUT-D refers to BDLUT trained with the extra two-stage noise degradation model for producing noisy data, which is removed during inference.

where x_i is input to a branch, LUT_k is the k^{th} LUT, R_j and R_j^{-1} denote the j^{th} rotation 90° operation and its inverse operation, N is the number of kernels, and M_k is the number of rotation operations for the corresponding kernel.

For image denoising, larger receptive field (RF) generally improves performance, as also demonstrated by Luo et al.[20] However, the storage size of the look-up table (LUT) can be mathematically expressed as v^n, where v denotes the quantization bit width and n represents the number of input. Consequently, the storage requirement exhibits exponential growth in proportion to the number of input pixels provided to the corresponding convolutional neural network (CNN). To enlarge the RF for better image noise reduction without excessively consuming hardware storage resources, we employ the rotation ensemble method, which is widely utilized in SR-LUT,[10] Mu-LUT,[11] and HK-LUT.[14] As detailed in Figure 4, the rotation ensemble method employs four distinct 3-pixel kernels to achieve full coverage of the 5×5 receptive field (RF) region. Each kernel is systematically rotated around the pivot point through four angular configurations: 0°, 90°, 180°, and 270°, with the combined set ensuring comprehensive spatial sampling. All kernels overlap at the pivot point, and the L kernel overlaps with the H kernel area. This method enables the use of substantially larger RF while limiting CNN input pixel count.

In the process of implementing our model in hardware, the primary operations involved are addition, division, and LUT. It is widely understood that the division operation consumes the most resources among all operations when implemented in hardware. In our BDLUT model, we change the value of N/M_k in Equation (1) from 3/4, as in the MSB branch of HKLUT model, to 4/4. This modification is achieved by adding an L-kernel, which effectively makes the two divisors even. This strategic enhancement optimizes the division operation in our model, transforming it into a shift operation. Furthermore, within the RF area of the CNN, pixels at the center have a greater influence on the model's output results, as demonstrated by Luo et al.[20] Hence, we designated the fourth kernel as the L-kernel (as shown in Figure 4). This design allows for a more meticulous learning of the characteristics of the RF central area in the image. This is built upon the foundational work of the first three kernels (H, D, and B), which ensure an RF size of 5×5. Overall, this strategic enhancement not only makes the model highly efficient when implemented in FPGA but also significantly improves its image denoising performance compared to other LUT method.

Certainly, the incorporation of an L-kernel to form the BDLUT does marginally escalate the model size. However, this slight increase is inconsequential when compared to the substantial savings in hardware resources and the improvement in image denoising performance. In fact, even after this modification, our model size remains the smallest compared to other widely used LUT models for image processing. In Table 6, after integrating the L kernel into both the MSB and LSB branches, the PSNR of the image improved.

In the pursuit of enhanced denoising efficacy and hardware compatibility, we have also modified our model to employ a symmetrical parallel structure. Specifically, we set the number of kernels in the two branches MSB/LSB to 4/4. The image denoising capability can be optimized by expanding the RF of both the MSB and LSB of the input pixels. Our subsequent experimental results substantiate that this symmetrical parallel structure offers the best image denoising proficiency (cf. Table 6). Moreover, despite the increase in model size, the hardware memory remains entirely adequate for its accommodation.

3.2 | Real-world blind noise degradation model

Existing LUT methods, whether employed for image super-resolution or denoising tasks, are specifically trained for certain application scenarios with known resolution scales or noise levels. However, in real-world

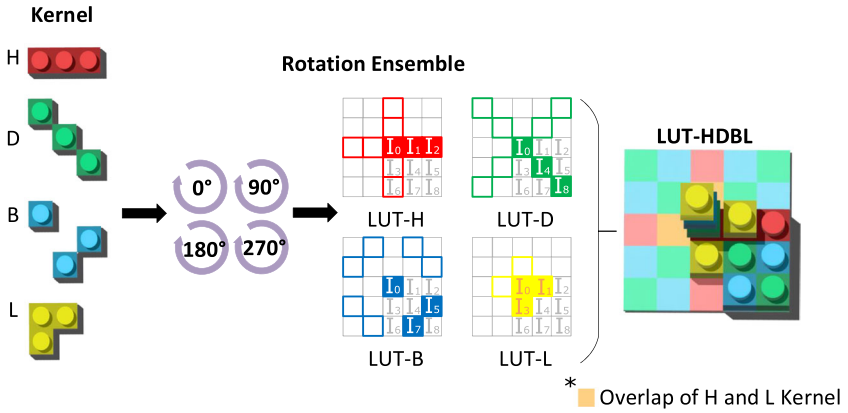


FIGURE 4 LUTs with four different kernels are on the left. Colored-bordered squares indicate pixel positions obtainable through 90° rotations. On the right, the final LUT composition displays squares with multiple colors, representing pixels extracted by multiple LUTs, fully covering a 5×5 RF.

applications, the type and intensity of noise are often unknown and complex. To enhance the generality and applicability of the model, we integrate a two-stage stochastic noise degradation model, adapted from Real-ESRGAN,[9] into our training process, as illustrated in Figure 2. This modification aims to simulate unknown or real-world noise, thereby aiding our BDLUT in achieving superior denoising results on blind and real-world noise.

In the degradation model, we first chose Gaussian additive noise. It is well known that the Gaussian noise assumption is the most prudent choice in the absence of noise information.[21] Traditional degradation models[16,17] often use Gaussian additive noise to simulate real-world noise, but this alone is insufficient. To better reflect real-world conditions, we add Poisson noise as a complementary noise type. Gaussian additive noise follows a Gaussian distribution, with the intensity being dependent on the standard deviation of this distribution. By adding independently sampled noise to each channel of the RGB image, we can generate color noise. However, as demonstrated by Nam and Hwang et al.,[22] the camera imaging process typically removes the channel-independent characteristics of noise. To simulate this realistic condition, we apply the same sampled noise across all three channels, thereby generating gray noise. Poisson noise, on the other hand, adheres to a Poisson distribution, and its intensity escalates in proportion to the increase in image intensity.

Figure 3 illustrates the structure of the noise model and the process of its constituent BDLUT for blind image denoising. Our approach involves superimposing noise onto the image twice, with each instance's type of noise (Gaussian or Poisson) and method of addition (color or gray) randomly determined by probabilistic parameters. Parameters relating to noise intensity, among others, are also randomly chosen within a specific range. The classical degradation model[16,17] is limited to a fixed set of basic degradation types, representing a first-order approximation of real-world degradation processes. Recently, BSRGAN[18] introduces a random shuffling strategy to synthesize more realistic degradation patterns. Nevertheless, this approach remains constrained by a fixed set of degradation processes, and the practical utility of all shuffled degradations remains uncertain. In contrast, our second-order degradation modeling for the noise model provides enhanced flexibility and strikes an optimal balance between simplicity and effectiveness. Overall, our strategy aptly simulates the characteristics of blind noise or real-world diverse degradation processes in the absence of training data. Then we implement the two-stage noise degradation model using a carefully designed parameter set and train the LUT model using a simulated dataset generated by this noise model. The experimental results of our model processed through the degradation model outperform or approach the performance of some non-blind models.

In addition, the noise degradation model enhances the stability and generalization capability of BDLUT, enabling it to handle variable noise more effectively. As shown in Figure 1, BDLUT-D, which incorporates the degradation model, maintains a high PSNR on datasets containing multiple types of noise, demonstrating its robustness and generalizability. Furthermore, the non-blind model BDLUT also achieves competitive PSNR performance, indicating that the structural design of our model also contributes to its robustness.

4 | DESIGN OF THE HARDWARE LOOKUP TABLE MODULE

4.1 | Architecture of the lookup table module

In previous studies,[1,3,4] shardware implementations of noise reduction have utilized BF method due to its simplicity. However, the algorithm's reliance on multiplication necessitates the use of hardware multipliers, which consume significant resources and introduce delays. Furthermore, the denoising performance of the BF does not match that of LUT and DNN methods. Meanwhile, building on the hardware-friendly Light-DnCNN,[15] a CNN accelerator incorporating depth-wise separable convolution and a fused-layer architecture was designed. Despite extensive optimization of both the algorithm and hardware implementation, the numerous MAC operations inherent in CNN algorithms inevitably consume substantial logic resources. To address these challenges, several previous studies have proposed LUT-based super-resolution and denoising methods, highlighting their hardware advantages and conducting experiments on CPUs or Raspberry Pi platforms. However, none of these studies have mapped the LUT method to FPGA or ASIC implementations.

This work presents the first FPGA implementation of a LUT processing element (PE) by mapping the table lookup module of BDLUT to hardware. The overall architecture is shown in Figure 5. BDLUT employs four types of kernels—H, D, B, and L—each covering a 5×5 receptive field, rotated by 0°, 90°, 180°, and 270°, respectively. Thus, our LUT PE accepts a 5×5 RGB pixel block as input.

To enhance data processing parallelism and accelerate video stream processing, the 5×5 RGB pixel block is split into its R, G, and B channels. Each channel is then processed in parallel by three identical LUT calculation units. After processing, each channel outputs a pixel

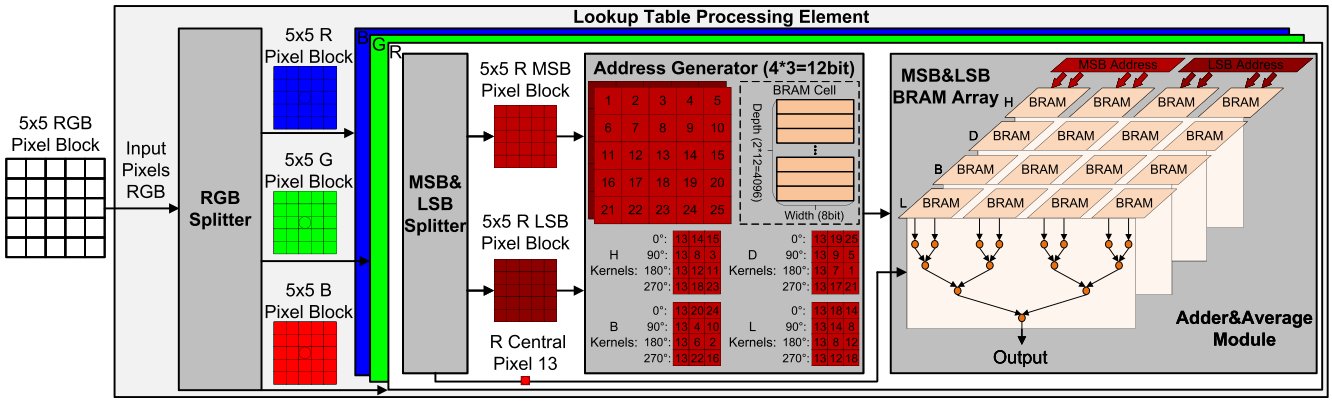


FIGURE 5 Architecture of the LUT processing element (PE), showing the RGB channel separation, parallel processing paths, and efficient memory organization.

corresponding to the central pixel of the 5×5 block, thereby performing noise reduction. Finally, the processed R, G, and B channels are reassembled into a single 3-channel RGB pixel.

4.2 | Circuit implementation

The hardware implementation across the three channels is consistent, as shown in Figure 5. For each channel, a 5×5 pixel block (comprising 8-bit pixels) is first divided into most significant bit (MSB, 4-bit) and least significant bit (LSB, 4-bit) components. This separation enables differential processing based on bit significance. Concurrently, the center pixel is extracted for subsequent addition.

In each kernel, the algorithmic advantages are reflected in the hardware design, allowing the Block RAM (BRAM) size to be reduced from 2^8 to 2×2^4. Both the MSB and LSB pixel blocks are fed into the address generator. Based on the arrangement of the H, D, B, and L kernels within the 5×5 pixel grid, a 8-bit or 12-bit address is formed for every three pixels. Additionally, since the algorithm includes image rotations of 0°, 90°, 180°, and 270°, the address generator produces a total of 32 addresses, which are sent to the BRAM Array. Meantime, to minimize the overall storage size, each individual BRAM cell uses dual-port BRAM, effectively halving the total BRAM resource usage. The BRAM Array consists of 16 dual-port BRAM cells, each with a depth of 2^8 for LSB and 2^{12} for MSB and a width of 8. In each cycle, 32 8-bit values are retrieved simultaneously. These 32 values, along with the center pixel generated by the MSB and LSB splitter, enter the adder tree module for averaging. The averaging process is performed by a shift module that shifts the result 1 or 2 bits to the right.

To mitigate shift errors caused by the MSB and LSB during shifting and accumulation, a selection judgment is made on the last 2-bit value before the shift. This determines whether to add 1 to the final result, ensuring that the hardware accurately maps the algorithm without introducing errors.

In the overall LUT PE, the entire calculation module excludes multiplication operations, significantly reducing logic resource consumption and processing delays.

5 | EXPERIMENTS

5.1 | Experiment setup

5.1.1 | Datasets and metrics

We use DIV2K[23] dataset as training set. For testing the denoising effect of BDLUT with known noise level, we add additive White Gaussian noise (AWGN) with three noise levels, that is, $\sigma = 15, 25, 50$ to DIV2 K to train the model. We also use DIV2 K with the addition of random noise generated by the noise degradation model to train a model BDLUT-D specialized for blind image denoising, and test under blind AWGN and unknown real-world noise. In the noise degradation model, we add Gaussian noises and Poisson noises to image with a probability of 0.5 and 0.5, respectively. The noise sigma range and Poisson noise scale are chosen between [0,50] and [0.05,3], respectively ([0,50] and [0.05, 0.25] for the second-order noise stage). The gray noise probability is set to 0.4 for both two stage noise addition. We use widely used peak signal-to-noise ratio (PSNR) as an evaluation metric. For color image denosing, CBSD68,[24] Kodak24,[25] McMaster,[26] and Urban100[27] are used as the benchmark dataset. For grayscale image denosing, BSD68[24] and Set12[8] are used as the benchmark dataset. For real-world noise denosing, SIDD-Medium Dataset[28] is used as training dataset, and SIDD Validation Data[28] is used as benchmark dataset. To simulate realistic noise variations, we implement a composite evaluation protocol: From each

AWGN-corrupted test set (σ = 15, 25, 50), we strategically sample one-third of images to construct mixed-noise-intensity benchmarks. This systematic approach enables comprehensive assessment of model robustness across heterogeneous degradation levels, particularly revealing performance characteristics under non-uniform noise distributions. To further validate the robustness and generalization of BDLUT-D, and to conduct noise model ablation studies, we add salt and pepper (SP) noise to the benchmark dataset for testing.

5.1.2 | Experimental setting

The network is first trained with DIV2 K training images for 200 k iterations with a batch size of 16 on Nvidia RTX 3090 GPUs. The Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 1e-8$) with the MSE loss is used. The initial learning rate is set to 5×10^{-4}, which decays to one-tenth after 100 k and 150 k iterations, respectively. Then the network will be transformed into LUT for performance evaluation. Figure 6 illustrates the architecture and implementation of the LUT-based video streaming system. As shown in Figure 6A, the LUT PE was implemented on the ZCU102 FPGA platform. The algorithmic lookup table was first converted to the COE format and preloaded into BRAM. Output pixels were computed through the BRAM-based LUT and rigorously compared with algorithmic results to

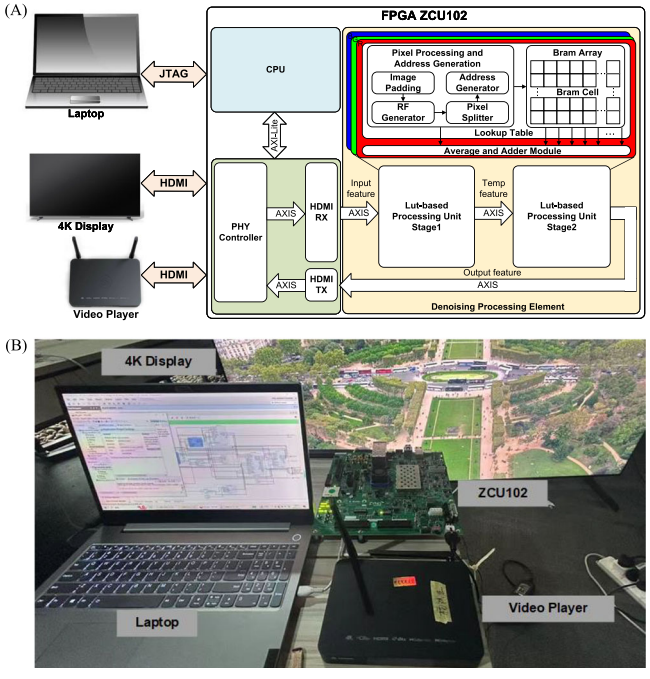


FIGURE 6 (A) Architecture of the LUT-based video streaming system implemented on the ZCU102 FPGA platform; (B) experimental setup for the FPGA-HDMI-CPU heterogeneous system.

TABLE 1 Size and quantitative comparison (PSNR/dB) for color image denoising on 4 fixed-noise-intensity benchmark datasets.

Category	Method	Size	CBSD68			Kodak24			Urban100			McMaster		
			$\sigma = 15$	$\sigma = 25$	$\sigma = 50$	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$
LUT-based	SR-LUT[10]	82 KB	32.72	29.60	25.41	33.44	30.31	25.98	32.42	28.69	24.05	34.69	31.03	26.11
	MuLUT-SDY-X2[11]	490 KB	33.20	30.48	27.12	34.01	31.32	28.04	33.23	30.00	25.84	35.56	32.77	29.18
	SPF-LUT-Net[12]	618 KB	34.02	31.30	28.03	34.85	32.23	29.02	34.49	31.30	27.29	36.52	33.78	30.25
	BDLUT-D (ours)	**66 KB**	31.46	30.52	25.02	32.05	31.19	25.88	30.62	29.71	25.30	34.08	32.80	26.60
	BDLUT (ours)	**66 KB**	33.41	30.56	27.25	34.18	31.42	27.93	33.1	29.92	26.2	35.67	32.84	29.12
Classical	CBM3D[5]	—	33.52	30.71	27.38	34.28	31.68	28.46	33.92	31.35	27.94	34.06	31.66	28.51
	MC-WNNM[6]	—	29.61	26.73	23.18	30.04	26.34	21.18	30.06	27.01	23.01	30.65	27.74	23.96
DNN	DnCNN[8]	2.65 MB	33.89	31.23	28.01	34.48	32.03	28.85	32.98	30.81	27.59	33.45	31.52	28.62
	SwinIR[7]	117 MB	34.42	31.78	28.56	35.34	32.89	29.79	35.13	32.90	29.82	35.61	33.20	30.22

Note: Smallest size of each model is shown in bold.

validate functional correctness. The image processing pipeline comprises a data preprocessing module and two parallel processing stages, enabling simultaneous data handling. To achieve real-time video streaming and seamless display, the system leverages the ZCU102's HDMI PHY interface for external video stream input/output. Specifically, the HDMI RX subsystem captures raw video streams via the HDMI PHY, converts them into the AXI-Stream (AXIS) format, and feeds them to the pixel processing modules. Conversely, the HDMI TX subsystem transmits processed AXIS data back to the HDMI PHY for output to a 4 K display. The entire HDMI IP framework is managed by the ZCU102's embedded Arm® Cortex®-A53 CPU. For video input, a Zidoo Z9X PRO player was utilized, supporting resolutions ranging from 480 P to 4 K. Figure 6B details the practical experimental setup. The FPGA development board interfaces with a laptop via JTAG and UART for bitstream programming and real-time debugging. The HDMI RX port connects to the video player via HDMI cable, while the HDMI TX port routes processed video data to the 4 K display. This configuration ensures end-to-end validation of the video processing pipeline under practical conditions.

5.1.3 | Baselines

Consistent with previous work, we evaluate BDLUT against several well-recognized denoising methods and LUT-based methods, including CBM3D,[5] MC-WNNM,[6] DnCNN,[8] SwinIR,[7] SR-LUT,[10] MuLUT,[11] and SPF-LUT.[12]

5.2 | Algorithm performance comparison of model accuracy

5.2.1 | Image denoising performance

Table 1 shows the performance of our method compared to various other methods for color and grayscale images with $\sigma = 15,25,50$ AWGN and also compare the size of different models. As evidenced by the table, our BDLUT framework achieves competitive denoising performance on color images, while maintaining a significantly smaller model size compared to other LUT-based methods-a clear demonstration of its architectural superiority. Notably, while the BDLUT-D variant (trained with randomized Gaussian noise levels) excels in blind AWGN denoising scenarios, its performance becomes suboptimal when tested on fixed-noise-level benchmarks. This limitation stems from the inherent mismatch between its noise-agnostic training protocol and the precisely defined noise characteristics in non-blind evaluation settings.

5.2.2 | General AWGN denoising performance

While BDLUT-D demonstrates a decrease in performance when evaluated on fixed-intensity Gaussian noise test sets (a result of its training with randomized noise levels), as shown in Tables 2 and 3, it significantly outperforms competing methods in mixed-noise-intensity benchmarks,

TABLE 2 Quantitative comparison (PSNR/dB) for colorscale image denoising on mixed-noise-intensity benchmark datasets that are constructed by evenly dividing the original dataset into three subsets, each corrupted with AWGN at $\sigma = 15$, 25, and 50, respectively.

Method	Train set (σ)	CBSD68	Kodak24	Urban100	McMaster
SR-LUT[10]	15	28.26	28.12	27.78	29.15
	25	28.13	28.70	27.44	29.71
	50	27.22	28.22	26.12	29.20
MuLUT-SDY-X2[11]	15	28.23	28.49	28.48	29.76
	25	28.11	28.72	27.95	30.15
	50	27.21	28.04	26.40	29.23
SPF-LUT-Net[12]	15	27.95	28.17	28.24	29.54
	25	27.94	28.41	28.47	30.27
	50	27.57	28.69	27.14	29.90
BDLUT (ours)	15	28.26	28.58	28.12	29.46
	25	28.46	29.19	28.17	30.49
	50	26.82	27.93	26.42	29.75
BDLUT-D (ours)	—	**29.00**	**29.70**	**28.54**	**31.16**

Note: Best performance (PSNR/dB) is shown in bold.

which better mimic the diverse noise environments encountered in real-world image degradation scenarios. This performance discrepancy arises from the inherent compatibility between BDLUT-D's noise-agnostic training approach and the varied, unpredictable noise characteristics typically found in practical imaging situations. This key advantage makes BDLUT-D particularly valuable for deployment in real-world applications, where noise conditions are often highly dynamic and cannot be easily anticipated or modeled in advance. In Table 4, the first three rows represent BDLUTs generated after training with images corrupted by AWGN at different noise levels, while the last row represents the BDLUT-D, which was trained using a specially designed degradation model. We evaluated these four models on the color benchmark dataset with AWGN at various noise levels. In the table, it is evident that BDLUT-D demonstrates greater robustness compared to BDLUT. While the model trained on a dataset with $\sigma = 15$ AWGN may perform well in handling images with $\sigma = 15$ AWGN, it may struggle when faced with new, unfamiliar image processing scenarios. Referencing Table 4, the BDLUT-D model ranks second only to the BDLUT model trained on a dataset degraded with a noise level matching that of the test dataset. Consequently, the integration of the degradation model in BDLUT-D enhances its cross-scene generalization capabilities. The noise degradation model, featuring adjustable parameters across a range, various noise types, and noise addition methods, may exhibit reduced PSNR when addressing a single noise type. However, this trade-off enhances its generality, resulting in overall improved performance when dealing with multiple noise types. Based on the observations above, it is evident that BDLUT-D offers distinct advantages in environments characterized by varying levels of image noise and a scarcity of suitable training datasets.

TABLE 3 Quantitative comparison (PSNR/dB) for grayscale image denoising on mixed-noise-intensity benchmark datasets that are constructed by evenly dividing the original dataset into three subsets, each corrupted with AWGN at $\sigma = 15$, 25, and 50, respectively.

Method	Train set (σ)	BSD68	Set12
SR-LUT[10]	15	23.85	24.17
	25	24.94	25.54
	50	24.88	25.64
MuLUT-SDY-X2[11]	15	24.24	24.67
	25	25.11	25.98
	50	25.21	25.97
SPF-LUT-Net[12]	15	23.85	24.25
	25	25.13	26.08
	50	25.74	26.94
BDLUT (ours)	15	24.21	24.56
	25	25.45	26.31
	50	25.48	26.41
BDLUT-D (ours)	—	**26.34**	**27.18**

Note: Best performance is shown in bold.

5.2.3 | Real-world noisy image denoising performance

Table 5 compares the denoising effect of each method on realistic noisy image dataset. While BDLUT-D may not perform as well as BDLUT in denoising realistic noise images from SIDD dataset, it offers greater practicality. In real-world settings, noise characteristics are often

TABLE 4 Quantitative comparison (PSNR/dB) for general AWGN image denoising.

		CBSD68			Kodak24			Urban100			McMaster		
Method	Train set	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$
BDLUT	$\sigma = 15$	33.41	29.68	21.70	34.18	29.91	21.65	33.17	29.39	21.79	35.67	30.62	22.09
	$\sigma = 25$	30.70	30.56	24.13	31.70	31.42	24.46	30.41	29.92	24.19	33.60	32.84	25.04
	$\sigma = 50$	26.78	26.88	26.81	28.73	28.63	27.93	26.52	26.47	26.27	30.09	30.04	29.12
BDLUT-D	—	31.46	30.52	25.02	32.05	31.19	25.88	30.62	29.71	25.30	34.08	32.80	26.60

Note: Best performance in each column is shown in blue, and the second best is shown in red.

TABLE 5 Quantitative comparison (PSNR/dB) for real-world image denoising on validation datasets.

	Method				
Dataset	SRLUT[10]	MuLUT-SDY-X2[11]	SPF-LUT-Net[12]	BDLUT-D	BDLUT
SIDD	34.30	40.09	40.20	36.44	39.37

unknown, and it is challenging to have a deterministic dataset covering the right mix of noisy data for robust training purpose. By incorporating the degradation model, BDLUT-D can be blind trained to effectively adapt to various real-world noise scenarios for image denoising. BDLUT excels in handling real-world image noise present in the training dataset but struggles with unlearned situations. In contrast, BDLUT-D demonstrates greater versatility and performs more effectively across a range of scenarios.

5.2.4 | Kernel and RF performance

Table 6 illustrates the color image denoising performance of BDLUT using various kernel and RF combinations on MSB and LSB. First, when comparing the selection of HD/HDB and HDB/HD for MSB/LSB, respectively, it is evident that HDB/HD is not uniformly superior to HD/HDB. The addition of sigma = 25, 50 AWGN for the McMaster dataset results in a larger PSNR for HD/HDB than for HDB/HD. And in other situations, the difference between HDB/HD and HD/HDB is not significant, with the PSNR difference remaining within 0.2 dB. Thus, both MSB and LSB make equal contributions to image denoising performance. Second, HDBL/HDBL surpasses HD/HD, HDB/HDB, HDB/HD, HDBL/HD, and HDBL/HD, indicating that a larger RF in either MSB or LSB branches corresponds to higher PSNR (i.e., better denoising result). Third, the PSNR of HDBL/HDBL is larger than that of HDB/HDB in denoising, signifying that the L-kernel LUT added to both the MSB and LSB branches is highly effective. Lastly, HDBL/HDBL has a higher PSNR than HD/HD, HDBL/HD, and HD/HDBL in image denoising tasks, indicating it is the best combination among all the hardware-friendly model combinations with an even number of kernels. Regarding model size, although HDBL/HDBL is the largest among all the kernel combinations, it remains relatively small and acceptable for hardware implementation.

5.2.5 | Qualitative analysis

Figures 7 and 8 visually demonstrate the denoising effects of various LUT methods, presenting comparative results for two distinct cases: color images with AWGN at a noise level of 15, and color images corrupted by real-world noise. As shown in Figures 7 and 8, while SR-LUT and MuLUT retain some noise artifacts in their denoised images and SPF-LUT produces blurred results with lost details and textures, our proposed BDLUT maintains a

TABLE 6 Quantitative comparison (PSNR/dB) for color image denoising on benchmark datasets with different kernel and RF size for MSB and LSB branch of our method.

Method	RF	Size	CBSD68			Kodak24			Urban100			McMaster		
MSB/LSB	MSB/LSB	(KB)	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$
HD/HD	3 × 3/3 × 3	3	32.76	29.57	26.42	33.42	30.14	26.96	31.84	28.46	24.75	34.19	31.63	28.16
HD/HDB	3 × 3/5 × 5	26.2	32.94	29.96	26.47	33.60	30.62	27.02	32.46	28.97	24.69	35.00	32.03	28.26
HDB/HD	5 × 5/3 × 3	26.2	33.09	30.01	26.52	33.82	30.79	27.17	32.64	28.97	24.97	35.28	31.49	27.77
HD/HDBL	3 × 3/5 × 5	34.5	32.87	29.90	26.34	33.45	30.55	26.93	32.27	28.83	24.39	34.75	31.96	28.14
HDBL/HD	5 × 5/3 × 3	34.5	33.33	30.31	26.82	34.07	31.10	27.40	33.09	29.57	25.32	35.46	32.52	28.38
HDB/HDB	5 × 5/5 × 5	49.5	33.35	30.50	26.96	34.11	31.32	27.59	33.04	29.86	25.43	35.66	32.74	28.71
HDBL/HDBL	5 × 5/5 × 5	66	**33.41**	**30.56**	**27.25**	**34.18**	**31.42**	**27.93**	**33.17**	**29.92**	**25.94**	**35.67**	**32.84**	**29.12**

Note: Best performance is shown in bold.

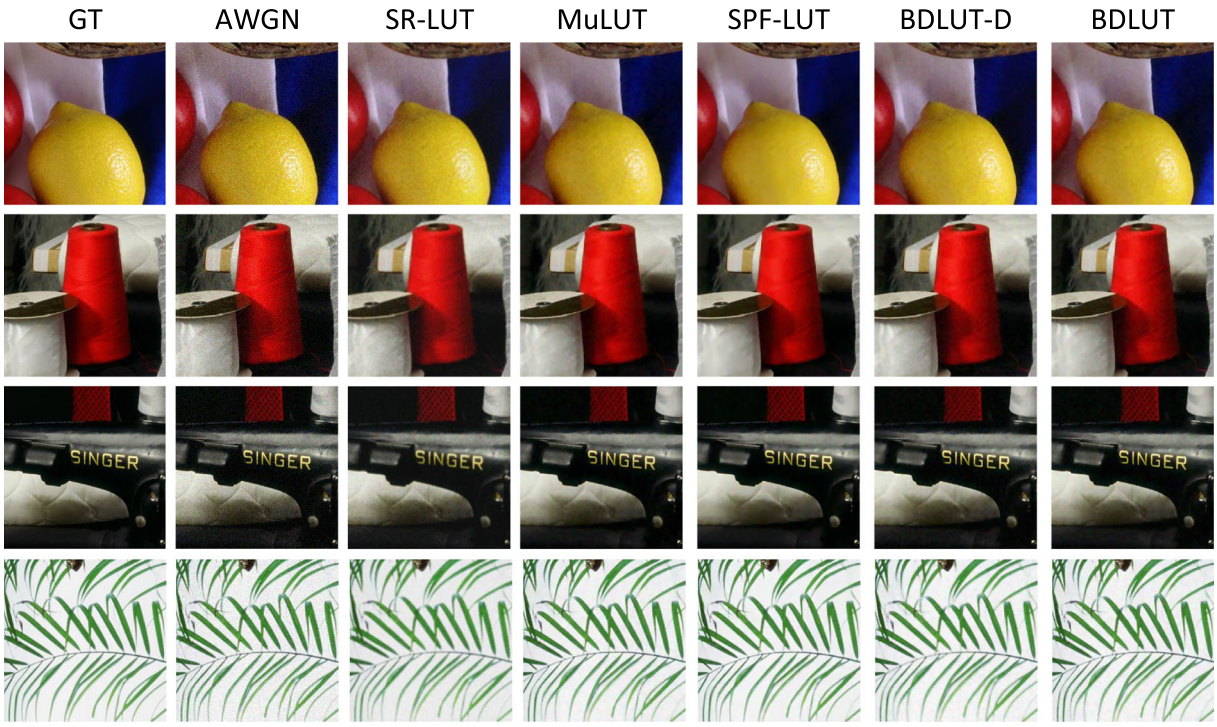


FIGURE 7 Qualitative comparison for color image denoising at a noise level of 15 AWGN on McMaster dataset.

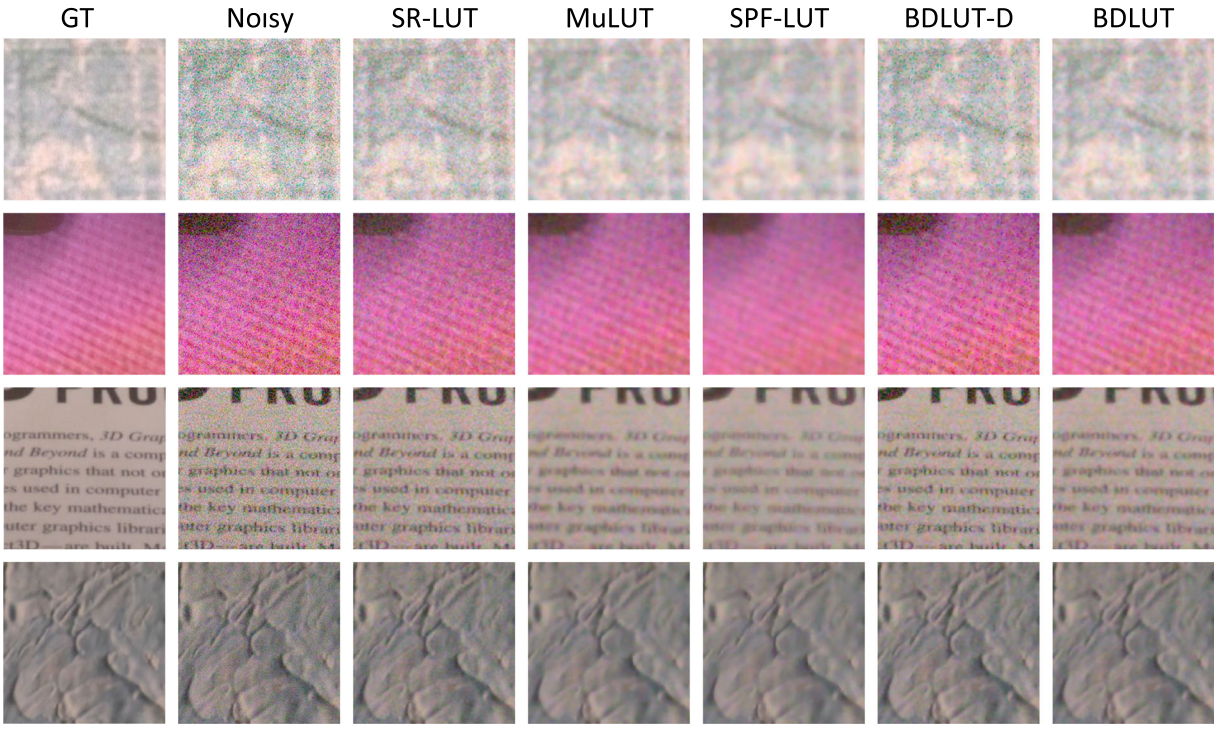


FIGURE 8 Qualitative comparison for color image denoising at real-world noise on SIDD validation dataset.

better balance between noise removal and detail retention. Furthermore, our blind denoising model BDLUT-D demonstrates competitive performance relative to other LUT methods, even without specific scenario training, suggesting its practical adaptability.

5.2.6 | Ablation study

Table 7 presents a comparative evaluation of BDLUT-D, incorporating the degradation model, against other LUT-based methods for handling unknown noise types

Method	Train set (σ)	CBSD68	Kodak24	Urban100	McMaster
SR-LUT[10]	15	18.37	18.71	19.45	19.64
	25	18.46	18.87	19.41	19.76
	50	18.66	19.12	19.31	20.09
MuLUT-SDY-X2[11]	15	24.88	25.14	24.82	24.95
	25	25.89	26.38	25.75	26.80
	50	26.00	27.06	24.99	27.34
SPF-LUT-Net[12]	15	24.37	24.61	24.29	24.30
	25	25.65	26.22	25.31	25.63
	50	26.96	28.15	26.32	28.51
BDLUT (ours)	15	24.60	24.88	24.41	24.42
	25	25.65	26.25	25.17	25.91
	50	26.96	27.83	25.66	28.61
BDLUT-D (ours)	—	**28.90**	**29.85**	**27.73**	**29.19**

TABLE 7 Quantitative comparison (PSNR/dB) for SP noise removal (best performance in bold).

Design	FPGA platform	Fmax (MHz)	Frame rate	Resource utilization			
				LUT	FF	BRAM (Kb)	DSP
[1]	XC7Z020	238	226.7	**1357**	2118	72	32
[3]	XC7Z020	250	52.45	1594	2399	**0**	6
[4]	Virtex-5	242	230	2529	1917	144	27
[15]	ZCU106	200	1.25	159967	29851	11232	736
Ours	ZCU102	**375**	**357.7**	7137	**162**	3026	**0**

TABLE 8 Post-implementation results for 1024×1024 images.

Note: Fastest frequency and frame rate and minimum resource consumption are shown in bold.

(e.g., SP noise commonly encountered in the display domain). The experimental results demonstrate that BDLUT-D achieves the highest PSNR values, indicating its superior generalizability in image denoising even when processing out-of-distribution noise patterns. These findings suggest that the integration of the degradation model significantly enhances BDLUT-D's robustness and generalizability versus other approaches, making it highly practical for real-world scenarios characterized by random and variable noise conditions.

5.3 | System performance comparison of hardware metrics

The LUT processing element (PE), which eliminates multiplication operations, achieves an operating frequency of 375 MHz on FPGA. Its architecture utilizes 7137 look-up tables (LUTs), 162 flip-flops (FFs), and 3026 Kb of BRAM while consuming 0.644 W of power, and notably avoids digital signal processing (DSP) units entirely. To validate the advantages of the LUT PE, we benchmark it against the BF design[1,3,4] and a DNN accelerator[15] for 1024×1024 image processing, as summarized in Table 8. Compared to the BF design, our approach demonstrates superior SSIM/PSNR metrics while maintaining comparable LUT and FF utilization and reducing DSP consumption by 100%. By eschewing multipliers, the LUT PE achieves higher maximum frequencies (375 MHz vs. prior works) and frame rates, significantly boosting throughput. Furthermore, the simplified architecture enhances maintainability and scalability, permitting straightforward adaptation to larger input resolutions or diverse FPGA platforms without compromising flexibility.

6 | CONCLUSION

This paper has proposed BDLUT(-D), a blind denoising approach that integrates hardware-optimized lookup tables to deliver a standalone ASIC IP solution for efficient and effective image restoration. BDLUT(-D) achieves SOTA denoising performance while reducing hardware resource usage by orders of magnitude compared to prior solutions, eliminating the need for specialized NPU/DNN accelerators. This efficiency enables seamless deployment in real-time edge applications, including low-end webcam enhancement and

computationally demanding 4 K denoising. Experiments demonstrate that BDLUT-D not only surpasses existing methods on realistic noise simulation benchmarks—particularly mixed-noise-intensity benchmarks but also provides a scalable, cost-effective pathway for mass-market imaging devices requiring minimal power and silicon footprint.

ACKNOWLEDGMENTS

This work was supported in part by the Theme-based Research Scheme (TRS) project T45-701/22-R and in part by the General Research Fund (GRF) Project 17203224, of the Research Grants Council (RGC), Hong Kong SAR.

ORCID

Boyu Li https://orcid.org/0000-0001-9709-9673

REFERENCES

1. Spagnolo F, Corsonello P, Frustaci F, Perri S. Design of approximate bilateral filters for image denoising on FPGAs. IEEE Access. 2023;11: 1990–2000.
2. Tomasi C, Manduchi R. Bilateral filtering for gray and color images. *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*; 1998. p. 839–46. https://doi.org/10.1109/ICCV.1998.710815
3. Spagnolo F, Corsonello P, Frustaci F, Perri S. Kit: kernel isotropic transformation of bilateral filters for image denoising on fpga. *2024 34th International Conference on Field-Programmable Logic and Applications (FPL)*; 2024. p. 19–23. https://doi.org/10.1109/FPL64840.2024.00013
4. Yao R, Chen L, Dong P, Chen Z, An F. A compact hardware architecture for bilateral filter with the combination of approximate computing and look-up table. IEEE Trans Circ Syst II Express Briefs. 2022;69(7): 3324–8.
5. Dabov K, Foi A, Katkovnik V, Egiazarian K. Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space. *2007 IEEE International Conference on Image Processing*, vol. 1; 2007. p. I–313–I-316. https://doi.org/10.1109/ICIP.2007.4378954
6. Xu J, Zhang L, Zhang D, Feng X. Multi-channel weighted nuclear norm minimization for real color image denoising. *2017 IEEE International Conference on Computer Vision (ICCV)*; 2017. p. 1096–104. https://doi.org/10.1109/ICCV.2017.125
7. Liang J, Cao J, Sun G, Zhang K, Van Gool L, Timofte R. Swinir: image restoration using swin transformer. *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*; 2021. p. 1833–44. https://doi.org/10.1109/ICCVW54120.2021.00210
8. Zhang K, Zuo W, Chen Y, Meng D, Zhang L. Beyond a gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Trans Image Process. 2017;26(7): 3142–55.
9. Wang X, Xie L, Dong C, Shan Y. Real-ESRGAN: training real-world blind super-resolution with pure synthetic data. *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*; 2021. p. 1905–14. https://doi.org/10.1109/ICCVW54120.2021.00217
10. Jo Y, Joo Kim S. Practical single-image super-resolution using look-up table. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; 2021. p. 691–700. https://doi.org/10.1109/CVPR46437.2021.00075
11. Li J, Chen C, Cheng Z, Xiong Z. Mulut: cooperating multiple look-up tables for efficient image super-resolution. *Computer Vision – ECCV 2022*; 2022. p. 238–56. https://doi.org/10.1007/978-3-031-19797-0_14
12. Li Y, Li J, Xiong Z. Look-up table compression for efficient image restoration. *2024 IEEE/CVF conference on computer vision and pattern recognition (cvpr)*; 2024. p. 26016–25.
13. Liu G, Ding Y, Li M, Sun M, Wen X, Wang B. Reconstructed convolution module based look-up tables for efficient image super-resolution. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*; 2023. p. 12183–92. https://doi.org/10.1109/ICCV51070.2023.01122
14. Huang B, Li JCL, Ran J, Li B, Zhou J, Yu D, Wong N. Hundred-kilobyte lookup tables for efficient single-image super-resolution. *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*; 2024. p. 857–65. https://doi.org/10.24963/ijcai.2024/95
15. Duan X, Xie R, Han J. An energy-efficient image denoising accelerator with depth-wise separable convolution and fused-layer architecture. *2021 IEEE 14th International Conference on ASIC (ASICON)*; 2021. p. 1–4. https://doi.org/10.1109/ASICON52560.2021.9620485
16. Elad M, Feuer A. Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images. IEEE Trans Image Process. 1997;6(12): 1646–58.
17. Liu C, Sun D. On Bayesian adaptive video super resolution. IEEE Trans Pattern Anal Machine Intell. 2014;36(2): 346–60.
18. Zhang K, Liang J, Van Gool L, Timofte R. Designing a practical degradation model for deep blind image super-resolution. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*; 2021. p. 4771–80. https://doi.org/10.1109/ICCV48922.2021.00475
19. Ma C, Zhang J, Zhou J, Lu J. Learning series-parallel lookup tables for efficient image super-resolution. *Computer Vision – ECCV 2022*; 2022. p. 305–21. https://doi.org/10.1007/978-3-031-19790-1_19
20. Luo W, Li Y, Urtasun R, Zemel R. Understanding the effective receptive field in deep convolutional neural networks. *Proceedings of the 30th International Conference on Neural Information Processing Systems*; 2016. p. 4905–13. https://dl.acm.org/doi/10.5555/3157382.3157645
21. Park S, Serpedin E, Qaraqe K. Gaussian assumption: the least favorable but the most useful [lecture notes]. IEEE Sig Process Mag. 2013;30(3): 183–6.
22. Nam S, Hwang Y, Matsushita Y, Kim SJ. A holistic approach to cross-channel image noise modeling and its application to image denoising. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; 2016. p. 1683–91. https://doi.org/10.1109/CVPR.2016.186
23. Agustsson E, Timofte R. Ntire 2017 challenge on single image super-resolution: dataset and study. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*; 2017. p. 1122–31. https://doi.org/10.1109/CVPRW.2017.150
24. Martin D, Fowlkes C, Tal D, Malik J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2; 2001. p. 416–23. https://doi.org/10.1109/ICCV.2001.937655
25. Franzen R. Kodak lossless true color image suite; 1999.

26. Lei Z, Xiaolin W, Antoni B, Xin L. Color demosaicking by local directional interpolation and non-local adaptive thresholding. J Electron Imag. 2011;20(2): 23016.
27. Huang J-B, Singh A, Ahuja N. Single image super-resolution from transformed self-exemplars. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2; 2015. p. 5197–206. https://doi.org/10.1109/CVPR.2015.7299156
28. Abdelrahman A, Lin S, Brown MS. A high-quality denoising dataset for smartphone cameras. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*; 2018. p. 1692–700. https://doi.org/10.1109/CVPR.2018.00182

AUTHOR BIOGRAPHIES

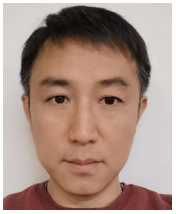
Boyu Li received the BS degree from the School of Microelectornics, Southern University of Science and Technology, Shenzhen, China, in 2022. He is currently pursuing the PhD degree with the Department of Electrical and Electronic Engineering, the University of Hong Kong, Hong Kong. His current research interests include deep learning, digital system design, and neural network acceleration.

Zhilin Ai received her BS degree from South China Normal University, China, in 2020, and MS degree from University of York, UK, in 2022. She is currently pursuing a PhD degree in Electrical and Electronic Engineering with the University of Hong Kong. Her current interests include edge AI, algorithm accelerator, and algorithm hardware co-design.

Baizhou Jiang received the BS degree in Electrical and Information Engineering from China University of Mining and Technology in 2022 and obtained the MS degree in Electrical and Electronic Engineering from the University of Hong Kong. His current research interests include image processing and embedded systems design.

Binxiao Huang received an MS degree in Beihang University in 2021. He is currently pursuing a PhD degree in department of electrical and electronic engineering from the University of Hong Kong. His research interests include efficient low-level image processing, 3D reconstruction, and multimodal generation.

Jason Chun Lok Li received the BEng degree from the Department of Electrical and Electronic Engineering, the University of Hong Kong, in 2020, and the PhD degree from the same department in 2024. His doctoral research focused on domain-specific techniques for efficient deep learning on edge devices. He is currently a Senior Research Engineer at the Audio Research Lab, Huawei Hong Kong Research Centre, where his research interests include neural audio coding and music generation.



Jie Liu received the MS degree in Electronic Information Engineering from the University of Science and Technology of China in 2006.He is currently a senior algorithm engineer at HongKong TCL Research Institute.

Zhengyuan Tu received his PhD in Condensed Matter Physics from Huazhong University of Science and Technology in 2019. He is currently working as a Senior FPGA Engineer at the TCL Industrial Research Institute. His current research interests include electrophoretic display and artificial intelligence.

Guoyu Wang graduated from Guangxi Normal University in 2016 with a master's degree in Electronic and Communication Engineering. He is currently an FPGA engineer at TCL Industrial Research Institute, and his research interests include image processing and artificial intelligence.

Daihai Yu (PhD in Computer Science, AI and computer vision) is holding the position on General Manager of TCL Corporate Research in Hong Kong. His research areas are AI, computer vision, pattern recognition, machine learning, and so forth. He has worked more than 10 years in international companies and leading AI research and development for different industries such as security surveillance,

AI home appliances, Smart home, smart manufacturing, AI display, and so forth.

Ngai Wong (Senior Member, IEEE) received the BEng and PhD degrees in electrical and electronic engineering from the University of Hong Kong (HKU), Hong Kong. He was a Visiting Scholar with Purdue University, West Lafayette, IN, USA, in 2003. He is currently an Associate Professor with the Department of Electrical and Electronic Engineering, HKU. He was the Associate Editor of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems from Jan 2014–June 2022 and has served as the track chair and member in the technical program committees (TPCs) of premier EDA conferences every year including DAC, ICCAD, and ASP-DAC. He co-founded the IEEE Council on EDA (CEDA), Hong Kong Chapter in 2016, and was the Chair in 2018/19. His research interests include compact neural network design, compute-in-memory (CIM) AI chips, electronic design automation (EDA), and tensor algebra.

How to cite this article: Li B, Ai Z, Jiang B, Huang B, Li JCL, Liu J, et al. BDLUT: Blind image denoising with hardware-optimized look-up tables. J Soc Inf Display. 2025;33(5):628–43. https://doi.org/10.1002/jsid.2075