# The progress of image denoising based on FPGA

Jiaye Yu *

School of Electronics and Information Engineering, Hangzhou Dianzi University, Hangzhou, China

* Corresponding Author Email: jayyu@hdu.edu.cn

**Abstract.** With the increasing demand for high-quality images in applications such as intelligent driving and medical imaging, the removal of noise like Gaussian and salt-and-pepper noise has become a significant challenge. Traditional software-based denoising methods, while flexible, often fail to meet real-time performance and high throughput requirements, especially with the growing scale of image data. FPGA (Field-Programmable Gate Array) provides a promising alternative with its high parallelism and low power consumption, enabling efficient hardware-level image processing. This paper aims to FPGA-based image-denoising algorithms, categorizing them into three main types: spatial domain, transformation domain, and deep learning methods. It explores their theoretical foundations, characteristics, and adaptability to FPGA platforms. In addition, optimization techniques such as data reuse, pipelining, and parallelization are discussed to improve resource efficiency and processing speed. Practical implementations highlight the advantages of FPGA in balancing computational demands and hardware constraints. While FPGA offers significant advantages, challenges remain, such as handling complex algorithms with limited resources and addressing high design complexity. The paper underscores the importance of combining hardware optimization with advanced algorithmic designs, including hybrid approaches with CPUs, to further accelerate processing. By addressing these challenges, FPGA-based solutions hold potential for advancing real-time image-denoising applications. This paper provides a comprehensive understanding of the current state of FPGA-based image denoising and serves as a reference for researchers and engineers seeking innovative solutions in this field.

**Keywords:** FPGA; image denoising; acceleration.

## 1. Introduction

Image processing, as a technology for re-encoding, processing, and analysing image information, has developed rapidly with the growth of computer technology. In modern industries, particularly in fields that rely on visual information, image processing plays a crucial role. In the field of image processing, noise comes from various sources. For example, the random thermal motion of electrons in electronic devices or external electromagnetic interference coupling into the system can introduce irregular interference signals into images. Common types of noise include Gaussian noise and salt-pepper noise, both of which appear as randomly distributed bright and dark spots on images while the former's noise amplitude distribution follows the Gaussian distribution, and the latter can be divided into black spots (pepper noise) and white spots (salt noise). Many fields demand high image quality. For instance, in intelligent driving, severely noisy images may lead to system misjudgements of the surrounding road conditions, affecting decision-making and safety. In biomedical imaging, excessive noise can obscure the signs of disease, hindering physicians' identifying lesions and potentially delaying diagnosis.

Traditional image processing algorithms are based on software programming languages (such as Python, and C++) and they are widely used due to their flexibility and massive tools support (such as OpenCV, and TensorFlow) [1-3]. However, with the sharp increase in required data throughput and the growing demand for real-time performance, software-based image processing methods have begun to face performance bottlenecks in many applications. In situations with high resolution, large data volume, and high real-time requirements, their processing speed and latency often fail to meet the demands. Therefore, FPGA (Field Programmable Gate Array), characterized by their highly parallel and low-power features, has emerged in recent years as a promising and effective hardware acceleration solution. The parallel architecture of FPGAs allows users to implement parallel image

processing operations directly at the hardware level, significantly improving processing speed and reducing power consumption. The most common FPGA-based image-denoising algorithms are spatial domain filtering techniques such as mean and median filtering. These have gradually evolved to include frequency domain algorithms such as wavelet transform-based methods. Additionally, with the rise of machine learning in recent years, lightweight deep learning-based denoising algorithms have also started to appear [1-3]. Beyond algorithmic advancements, researchers have proposed hardware optimization techniques for FPGAs, such as data reuse, pipelining, and parallelization, to achieve optimal denoising performance within the constraints of limited hardware resources.

This paper provides a review of FPGA-based image-denoising algorithms, classifying them into three categories based on their implementation architectures: spatial domain methods, transform domain methods and deep learning methods. It reviews the theories and methodologies of these algorithms, discusses their denoising effectiveness, and explores their adaptability on FPGA platforms. Finally, this review also explains how to achieve efficient hardware acceleration through structural optimization in FPGA implementations. This review aims to summarize and analyse the current research status of FPGA-based image-denoising algorithms, providing valuable references for researchers and engineers in related fields.

## 2. Image Denoising Algorithms Based on FPGAs

Image denoising algorithms can generally be categorized into three types based on their architectures: spatial domain methods, frequency domain methods, and deep learning methods. Among these, spatial domain-based algorithms are the earliest to understand and have undergone longer development, while deep learning-based denoising algorithms have become a major research focus in recent years due to the rapid advancements in the field of deep learning. In the following sections, this paper will introduce these three types of denoising algorithms in detail.

### 2.1. Spatial Domain Denoising Algorithms

Spatial domain denoising methods operate directly on the actual pixel of an image, achieving noise reduction by analysing the relationships between a pixel and its neighbouring pixels. Common algorithms in this category include mean filtering, median filtering, and non-local means filtering, which are relatively easy to understand, simple to implement, and computationally efficient. They often perform well in preserving the edges and details of an image. However, their effectiveness is usually noise-specific and hard to handle images with complex noise patterns. Algorithms based on the spatial domain often involve repetitive operations such as pixel window sliding, which are easily designed into pipeline systems and can fully leverage the parallel architecture of FPGAs to greatly improve denoising speed.

Spatial domain denoising algorithms have been developed for a long time, and in recent years, researchers have focused more on improving existing algorithms and proposing more efficient and widely applicable algorithms.

Lone M.R proposed a nearest neighbour filtering method (NNFM) based on Euclidean distance for removing impulse noise from images [4]. The classical approach for impulse noise is median filtering (MF), but its performance degrades sharply when noise density exceeds 50%. To address the limitations of median filtering, researchers have introduced various algorithms, such as Adaptive Median Filtering (AMF) and Decision-Based Algorithm (DBA). The NNFM algorithm first compares the grayscale value of each pixel with a preset threshold. If a pixel's value is close to the maximum or minimum grayscale value, it will be polluted by noise and marked as a noise pixel. Then, based on the continuity of image pixels, it uses Euclidean distance as the criterion to find the nearest non-noise pixel within a set domain to replace the noise pixel. The author achieved a processing rate of 271 frames per second for 720p resolution images using this algorithm. Moreover, when the noise density exceeds 20%, the denoising performance is superior to that of similar algorithms.

Spagnolo P. et al. proposed an approximate bilateral filter to remove Gaussian noise from images [5]. Bilateral filtering works by assigning weights based on spatial proximity and intensity similarity between pixels, performing denoising through weighted averaging. This method achieves good denoising results while preserving edges and texture details. However, its high computational complexity and power consumption pose challenges to efficient implementation of resource-constrained FPGAs. To address this, the authors introduced the concept of approximation. They approximated the coefficients of the spatial and intensity range kernels as unsigned integers and replaced the traditional nonlinear weighted averaging operation with a simple piecewise function. By optimizing the hardware architecture, they significantly reduced computational complexity. Although this approach compromises some denoising quality, the cleverly designed approximation strategy keeps the quality loss within visual acceptability. Using this algorithm, the authors achieved a processing speed of 926.8 frames per second for 512×512 grayscale images with a power consumption of only 63mW@244MHz. However, the method also has the limitations of bilateral filtering, such as tending to remove textures and details, flatting intensity regions, and potentially creating new contours. In addition, various improved bilateral filtering algorithms have been proposed, such as the adaptive bilateral filter by A. Xie et al. and the trilateral filter by Ramkumar Raja M et al. [6, 7].

### 2.2. Frequency Domain Denoising Algorithms

Frequency domain denoising algorithms work by transforming the image from the spatial domain to the frequency domain, processing the frequency spectrum to separate noise from the image information, and then converting it back to the spatial domain to get a clear image. These methods are particularly effective at removing periodic noise and common algorithms in this category include wavelet transform denoising and Fourier transform denoising. However, these algorithms tend to have higher computational complexity and are less effective for handling non-periodic noise. Additionally, the principles of these algorithms make them more difficult to implement on FPGA compared to spatial domain methods. As a result, there has been less research on FPGA-based frequency domain denoising algorithms in recent years compared to spatial or deep learning-based approaches.

However, there has been a notable amount of research on hybrid denoising algorithms that combine both spatial and frequency domains. These algorithms leverage the advantages of both domains. A representative algorithm in this category is the Block Matching 3D algorithm (BM3D).

Q. Huang et al. used the block-matching 3D filter to improve the quality of electromagnetic field patterns in device near-field scanning [8]. This algorithm assumes that the noise in the electromagnetic field pattern is zero-mean Gaussian distributed, described by variance to indicate the intensity of the noise. Then, based on Euclidean distance, it slides a set window to find similar pixel blocks. If the distance between two blocks is less than a threshold, these two blocks are considered similar. The similar blocks found are formed into a 3D matrix, which is then subjected to discrete cosine transform (DCT), transforming the block from the spatial domain to the transform domain. In the transform domain, noise is removed through threshold filtering and Wiener filtering, and the filtered 3D matrix is inversely transformed back to the spatial domain to obtain the denoised image. The authors used this algorithm to improve the clarity of electromagnetic field patterns, locate EMI sources, and pattern clustering, showing significant advantages, especially in processing weakly emitted electromagnetic patterns.

### 2.3. Deep Learning-Based Denoising Algorithms

Deep learning, as a rapidly developing field in recent years, has led to significant advancements in deep learning-based denoising algorithms. These algorithms utilize deep neural networks to learn the mapping relationship between noisy and clean images, achieving ideal denoising effects. As they can learn deep-level features of images, they have the powerful capabilities to recognize and handle complex noise patterns. However, this approach requires a large amount of training data and

computational resources, and the deployment of models, especially for real-time processing, presents a significant challenge. Representative algorithms include DnCNN and FFDNet.

Traditional Convolutional Neural Networks (CNNs) process pixels layer by layer, which, due to the need to store the complete intermediate feature maps, leads to significant I/O consumption. S. Coleman et al. proposed a depth-first (DF) CNN coprocessor architecture with high utilization and flexibility [9]. This architecture uses the output features from the previous convolutional layer to compute the output features of the next layer. The DF method includes three approaches: the Line-Based method, which only takes a new input row when the current row reaches the last layer; the Intermediate-Line method, which stores a full row of data for each intermediate layer, while other rows contain temporary results; and the Multiline method, which computes multiple rows at once. Additionally, the authors designed a parallelized DF approach, reducing I/O bottlenecks and improving throughput. They also introduced a programmable computing array for widespread deployment, maintaining high utilization of Digital Signal Processors (DSPs), and achieved a computation rate of up to 695 GOPS on the Zynq XCZU9EG board.

W. Zhao et al. proposed a new FPGA-compatible floating-point data format, called SFP [10]. The SFP format represents a set of floating-point numbers by sharing exponents and correction bits, effectively compressing the size of the floating-point data while maintaining good DNN model accuracy. It does not require fine-tuning of the network to adapt to quantization. The authors also proposed a neural network computation unit based on FPGA that is compatible with the SFP format, specializing in matrix multiplication, accumulation operations, and the accumulation of intermediate results produced by spatial multiply-accumulate modules, thereby improving the efficiency of SFP data and floating-point accumulation operations. To increase DDR bandwidth utilization, the authors designed a hardware-software co-scheduling scheme to balance the computational and memory bandwidth requirements of the neural network. Through a series of optimizations, this work achieved a computation rate of up to 4.072 TFLOPS on the Xilinx ZU9 FPGA.

## 3. FPGA Denoising Algorithm Structural Optimization Techniques

Although the parallel architecture of FPGAs can significantly enhance the processing speed of algorithms, their hardware characteristics make them less suited for floating-point operations and complex calculations such as multiplication, division, and recursion. Performing these operations directly on FPGA often consumes a large amount of hardware resources and requires the design of complex computation modules. Additionally, FPGA memory resources are relatively limited, making it difficult to load memory-intensive algorithms directly. Therefore, to run denoising algorithms more efficiently on hardware with limited resources, it is necessary to optimize the computational structure and process flow.

One major optimization method is vagueness and approximation. The former is primarily used in situations where similarity comparisons are needed, but the data accuracy is not crucial. They focus on the relative value, and objects can be fuzzily replaced to reduce computational complexity. The latter method focuses on optimizing complex function calculations on FPGA, making them more suited to FPGA hardware architectures. By approximating certain operations or simplifying calculations, it helps to align the algorithm with the hardware's strengths, thereby improving efficiency and reducing resource consumption.

Lone M.R et al. used a nearest neighbour filtering method based on Euclidean distance to remove impulse noise on FPGAs [4]. To avoid the multiplication and square root operations used in calculating Euclidean distance, they used a comparative and fuzzy thinking approach, using the absolute difference between pixel values (a simplified form of Manhattan distance) instead of Euclidean distance to determine the nearest domain, with the specific formula being:

$$|x_1 - y_1| + |x_2 - y_2| \tag{1}$$

Spagnolo P et al. employed an approximate method for bilateral filtering by approximating values to multi-bit unsigned integers [5]. This approach simplifies the computation of the most critical and complex components of the bilateral filter, namely the spatial kernel and intensity kernel while keeping the loss of denoising effectiveness within a reasonable range. This also facilitates subsequent operations.

Similarly, in the context of improving bilateral filtering, Bhargava G.U. and others proposed a modified recursive box filter (MRBF), which uses a polynomial approximation to replace traditional exponential operations, making it more efficient for hardware implementation [11].

Apart from vagueness and approximation, another major direction is improving the pixel scanning method.

For example, non-local mean filtering is an effective but computationally complex algorithm. In a traditional software implementation, designers propose to reuse as much computation data as possible to reduce redundant computations because this method requires retaining multiple columns or rows of data during scanning, which increases memory demand while improving computational speed. When this method is transplanted to FPGAs, under the high-speed parallel operation of FPGAs, memory limitations become a larger issue. Thus, as designed by Koizumi et al. [12], the scope of reused data can be divided, sacrificing computational speed for lower memory demand. Lone M.R et al. [4] divided the pixels within the filtering window into multiple neighbourhood layers from the centre to the periphery according to their distance, starting the search from the nearest neighbourhood layer and stopping the calculation as soon as the first non-noise pixel is found, reducing unnecessary comparisons, which is also an improvement on the scanning method.

## 4. Conclusion

This paper reviews the current state of image-denoising algorithms based on FPGAs, discussing the application and optimization methods of denoising algorithms in three directions: spatial domain, transform domain, and deep learning methods. Spatial domain algorithms such as mean filtering and median filtering are simple and can preserve image edges and details but have limited processing capabilities for complex noise. Transform domain algorithms such as wavelet transform denoising are suitable for periodic noise but have a large computational load, and the difficulty of implementing them on FPGAs is relatively high. Deep learning methods, by learning the mapping relationship between noise and clean images, have strong capabilities for processing complex noise patterns but require a large amount of data and computational resources, and the deployment and real-time performance of the model are challenges. At the same time, methods such as fuzzification, approximation, and scanning improvements can help denoising algorithms run efficiently on FPGAs. There are mainly two current challenges faced by image denoising algorithms based on FPGA: the first is the complexity of algorithms and the limitations of hardware resources, FPGA hardware resources are limited, not good at handling floating-point operations and complex function operations, and memory resources are scarce; the second is the high design difficulty and development cycle, the design difficulty and development cycle of FPGAs are high, which is not conducive to the design of complex algorithms. To address the above challenges, in addition to the improvements and optimization of algorithm structures mentioned in this paper, a combination of hardware and software can be used, such as working with CPUs to leverage the high computational speed of FPGAs and CPUs' abilities to handle complex operations to accelerate denoising algorithms. For the second challenge, HLS, automated design tools, and hardware-software co-simulation can be designed to reduce the development difficulty of FPGA denoising algorithms.

## References

[1] Wang M, Wang S, Ju X, Wang Y. Image Denoising Method Relying on Iterative Adaptive Weight-Mean Filtering. Symmetry, 2023, 15(6): 1181.

[2] Li, Y., Zhang, J. and Wang, M. Improved BM3D denoising method. IET Image Processing, 2011, 11: 1197-1204

[3] Bhawna G., Ayush D., Sunil A., B.S. Sohi, Apoorav S. Image denoising review: From classical to state-of-the-art approaches. Information Fusion, 2022, 55: 220-244.

[4] Lone M.R. An FPGA Implementation of an Impulse Noise Reduction Algorithm in Visual Sensor Network. Circuits Syst Signal Process, 2022, 41: 6136–6151.

[5] Spagnolo, P. Corsonello, F. Frustaci and S. Perri. Design of Approximate Bilateral Filters for Image Denoising on FPGAs. In IEEE Access, 2023, 11: 1990-2000.

[6] A. Xie, A. Zhang, G. Mei. An FPGA-Based Hardware Architecture of Gaussian-Adaptive Bilateral Filter for Real-Time Image Denoising.in IEEE Access, 2024, 12: 115277-115285.

[7] Ramkumar Raja M, Naveen R, Palaniswamy T, Mahendiran T, Shukla NK, Verma R. A field programmable gate array-based biomedical noise reduction framework using advanced trilateral filter. Transactions of the Institute of Measurement and Control, 2021, 43(16): 3588-3605.

[8] Q. Huang et al. Nonlocal Noise Removal Technique to Improve the Electromagnetic Field Pattern in Near-Field Scanning of a Device. in IEEE Sensors Journal, 2023, 23(6): 5901-5910.

[9] S. Colleman, M. Verhelst. High-Utilization, High-Flexibility Depth-First CNN Coprocessor for Image Pixel Processing on FPGA. in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2021, 29(3): 461-471.

[10] W. Zhao, Q. Dang, T. Xia, J. Zhang, N. Zheng and P. Ren. Optimizing FPGA-Based DNN Accelerator With Shared Exponential Floating-Point Format. in IEEE Transactions on Circuits and Systems I: Regular Papers, 2021, 70(11): 4478-4491.

[11] Bhargava, G.U., Gangadharan, S.V. FPGA Implementation of Modified Recursive Box Filter-Based Fast Bilateral Filter for Image Denoising. Circuits Syst Signal Process 2021,40: 1438–1457.

[12] Koizumi, Hayato & Maruyama, Tsutomu. An Implementation of Non-Local Means Algorithm on FPGA. Advances in Parallel Computing, 2020, 36: 681-690.