International Journal of Computer Science and Information Technology
ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 5, Number 1, Year 2025
DOI: https://doi.org/10.62051/ijcsit.v5n1.07
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Research on Hardware Acceleration of Dehazing Network Based on Deep Learning

Xianlong Zheng *

College of Electronic Information, Southwest Minzu University, Chengdu, China

ABSTRACT

Computer vision devices play a crucial role in both industrial and everyday life. However, when capturing images in hazy weather, the performance of these devices can be severely affected, resulting in a significant decrease in image quality. Therefore, effective image dehazing has become a necessary and urgent task. In recent years, with the rapid development of deep learning technology, significant progress has also been made in the field of image dehazing. However, despite the excellent performance of existing models in improving dehazing effects, they often overlook issues of computational efficiency and resource consumption. The high computational cost not only limits the widespread application of these advanced methods in practical scenarios, but also increases the difficulty and cost of hardware deployment. In response to the above challenges, we propose a novel lightweight image dehazing network called LDM-Net (Lightweight Dehazing Module Network). This network aims to reduce model complexity by optimizing structural design, while introducing attention mechanisms to enhance its generalization ability, thus better adapting to image dehazing needs in different environments. In order to verify the actual effectiveness and potential value of LDM-Net, we successfully deployed this network on various hardware platforms and conducted comprehensive testing and evaluation. The experimental results show that compared to several leading image dehazing solutions in the current market, LDM-Net can achieve better or at least equivalent dehazing performance while maintaining lower computational resource consumption. Specifically, it can approach or even surpass the performance of other advanced lightweight image dehazing methods in multiple key indicators, including but not limited to peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and visual perception evaluation. This achievement not only demonstrates the strong competitiveness of LDM-Net as an efficient and low-energy solution, but also points out a new direction for the research and development of image dehazing technology in the future. In addition, considering its good scalability and flexibility, LDM-Net is expected to be widely used in smart city construction, auto drive system and other fields, further promoting the progress and development of related industries.

KEYWORDS

Image dehazing; Deep learning; Lightweight network; Attention mechanism; Hardware deployment

1. INTRODUCTION

In daily life, due to industrial emissions, automobile exhaust, and other human activities, the concentration of particulate matter and water droplets in the atmosphere continues to increase [1], leading to a significant decrease in the quality of outdoor scene images. These turbid media, such as haze, scatter and absorb light, making distant objects blurry and reducing visibility, and have a negative impact on people's visual experience. In addition, for systems that rely on computer vision technology, the problem of image degradation caused by natural factors is particularly serious, as it directly affects the accuracy of algorithm recognition and overall performance [2]. To address this

issue, researchers have developed various dehazing techniques aimed at restoring the quality of photos or videos taken in polluted air. The main goal of dehazing is to improve image contrast, enhance detail clarity, and ultimately achieve the goal of improving scene visibility [3]. The process of image dehazing is shown in formula 1:

$$I(x) = J(x)t(x) + A(x)\big(1 - t(x)\big) \tag{1}$$

Among them, the foggy image represented by $I(x)$ is usually known. However, the haze free image represented by $J(x)$, the transmittance represented by $t(x)$, and the atmospheric illumination intensity represented by $A(x)$ are generally unknown. The application scope of image dehazing technology is wide, covering many fields such as image classification [4], traffic monitoring [5], medical imaging [6], image segmentation [7], driving assistance systems [8], and remote sensing monitoring [9].

With the development of science and technology, research on how to effectively remove fog interference in images has gradually deepened, forming a diversified solution system including methods based on physical models [10-12] methods that utilize prior knowledge [13, 14], and deep learning methods that have emerged in recent years. Among them, methods based on deep learning networks [15-17] have received widespread attention due to their powerful feature extraction capabilities and good generalization performance. However, such methods typically require high computational resource support, especially when dealing with large-scale datasets, which poses certain challenges for practical applications. Considering that the common central processing unit (CPU) on traditional embedded devices may not be able to meet high-performance requirements, using graphics processing units (GPU) or field programmable gate arrays (FPGA) as hardware accelerators has become a feasible choice. Especially FPGA, it not only provides sufficient computing power to perform complex image processing tasks, but also has low power consumption and flexible configuration capabilities, making it very suitable for applications that are cost sensitive and require high efficiency operation. Therefore, exploring the implementation of efficient dehazing algorithms based on FPGA platforms has become one of the current research hotspots. Overall, traditional image dehazing methods are based on both physical models and prior knowledge. Although the model can achieve good image dehazing effects in certain specific scenarios, its generalization ability is weak and there are certain limitations in its application scenarios. However, deep learning based dehazing methods, such as convolutional neural network (CNN) based methods [18-22], or Transformer based dehazing methods [23-27] although can effectively improve image dehazing performance and model generalization ability, there are shortcomings in resource consumption considerations, often requiring a large amount of computing resources, which is very unfriendly to hardware deployment and practical applications.

Therefore, in response to the above issues, this article proposes a system consisting of a lightweight image dehazing network and a corresponding hardware deployment framework. The system is based on a redefined Atmospheric Scattering Model (ASM) and is capable of learning both transmittance maps and atmospheric light intensity simultaneously. In order to compensate for the lack of generalization ability in lightweight image dehazing networks, this paper also introduces pixel attention mechanism (PA) to improve the learning performance of the network. The following are the main contributions of this article:

1) A system combining lightweight image dehazing network and hardware deployment framework is proposed to efficiently solve the problem of image dehazing.

2) Based on the redefined atmospheric scattering model, the joint learning of transmittance map and atmospheric illumination intensity has been achieved, enhancing the processing capability of the system.

3) By introducing pixel attention mechanism, the learning ability and generalization performance of lightweight image dehazing network have been effectively improved.

4) In order to objectively evaluate the performance of the proposed model and compare it with the latest advanced image dehazing algorithms, extensive experiments were conducted on multiple benchmark hazy image datasets, including synthesized indoor/outdoor and natural hazy images. The results indicate that the system proposed in this paper outperforms some of the best dehazing algorithms in terms of overall image dehazing performance.

The structure of the following chapters in this article is as follows: Chapter 2 will review the relevant research work in the field of image dehazing; Chapter 3 elaborates on the technical details and composition of the proposed method in detail; Chapter 4: In depth analysis and discussion of experimental results; Chapter 5 provides a summary of the entire text and offers prospects for future research directions.

2. RELATED WORK

This section aims to provide an overview of the advantages and limitations of various image dehazing algorithms, and to explore in depth the performance of these algorithms in hardware implementation, particularly on FPGA platforms. With the rapid development of image processing technology, image dehazing, as an important branch of computer vision, has attracted the attention of many scholars and given rise to various dehazing methods. These methods can generally be divided into two categories: one is based on traditional image processing techniques, and the other is based on modern algorithms of deep learning. Next, we will provide a detailed introduction to the advantages and disadvantages of these two types of algorithms, as well as the challenges and advantages they may encounter when deployed on FPGA.

2.1. Traditional Image Dehazing Algorithms

The dehazing algorithm based on image enhancement restores the details of blurry images through image enhancement techniques, improving visual quality, but may lose some detail information. Classic algorithms include histogram equalization [10], homomorphic filtering [11], Retinex [12], and other algorithms. Histogram equalization [10] uses statistical analysis of color grayscale distribution to nonlinearly stretch grayscale, increasing contrast and information content. However, this method does not pay attention to the original color, which may lead to color distortion and poor results. Dehazing algorithm based on homomorphic filtering: Homomorphic filtering also uses different algorithms to process different regions of the image in the frequency domain to reduce low-frequency noise interference such as haze, thereby improving image contrast. This algorithm equivalently transforms the problem of restoring foggy images into the problem of increasing the frequency domain range of foggy images. By reducing the low-frequency information in foggy images and correspondingly increasing the high-frequency information in the images, the effect of image dehazing can be achieved [28]. Overall, image enhancement based dehazing algorithms do not take into account the fundamental reasons for the decline in image quality in foggy weather. They only enhance contrast and saturation, and do not truly dehazing. Moreover, they sacrifice some image information and reduce image information entropy. Some of these algorithms have high complexity and are difficult to meet real-time dehazing requirements. It can be seen that the dehazing algorithm based on image enhancement is difficult to achieve significant breakthroughs. The defogging algorithm based on Retinex is based on the theory of color constancy [29], which suggests that objects can maintain their original color when the light source changes [28]. Retinex improves image contrast by removing lighting effects, and many researchers use this theory to restore the original color of objects under different light sources. However, this method is prone to producing halo phenomena, which can affect the defogging effect. The Retinex theory [30] also suggests that scene imaging is related to the light color around the object, and the human eye can distinguish the changes of the same object in different environments. Based on this, the Retinex algorithm eliminates uneven lighting and improves visual effects by estimating illuminance and decomposing reflectance images. The

proposed single scale Retinex (SSR) algorithm [31], multi-scale Retinex (MSR) algorithm [32], and color restoration multi-scale Retinex (MSRCR) algorithm [33] have all achieved good results.

Unlike image enhancement dehazing algorithms, image restoration dehazing algorithms focus more on the fundamental reasons for the impact of haze on image quality. They analyze the process of image quality degradation in the atmosphere from the perspective of atmospheric imaging physics models, and summarize and analyze several effective prior methods by observing a large number of foggy images. They estimate atmospheric parameters and scene color information [34], and then perform inverse operations based on the formation process of foggy images to restore clear images. Compared to the enhancement method, it achieved better dehazing effect. Tan et al. [35] proposed two hypotheses through statistical analysis: the contrast of foggy images is higher than that of foggy images, and atmospheric light changes smoothly in local areas. Based on these assumptions, they established a Markov random field model aimed at maximizing local contrast to achieve dehazing. However, this method is essentially contrast stretching, which may result in excessively high saturation of the restored image and halo effects in the edge regions. Fattal et al. [13] proposed a method for estimating light transmission in blurry scenes of a single image, redefining the atmospheric transmission model by treating each pixel in a haze free image as the product of the light reflectance factor and the reflected brightness. By projecting the three-dimensional color space vector of each pixel in the input image onto the atmospheric illumination vector and introducing surface shadows to eliminate scattered light, the contrast of foggy images can be restored, achieving a good dehazing effect. However, this model relies on the assumption of local independent components in foggy images, and the dehazing effect is limited for images with unclear changes in independent components such as dense fog. Tarel et al. [14] performed white balance processing on foggy images under the assumption of smooth changes in atmospheric light values, and used median filtering to calculate atmospheric light values. Due to the inability of median filtering to preserve edge information, the dehazed image may exhibit halo effects at changes in depth of field. He et al. [1] found through observation and statistics of a large number of outdoor fog free images that some pixels in non sky areas have at least one color channel with a low pixel value, indicating that fog makes dark pixels in degraded images brighter. Based on this prior, they estimated the medium transmission map and used an atmospheric scattering model to restore the fog free image. However, this method performs poorly and requires a large amount of computation when dealing with sky regions and high-intensity objects such as snow capped mountains and white clouds.

2.2. Deep Learning Based Image Dehazing Algorithms

The focus of early research on image dehazing based on deep learning is still to obtain more refined atmospheric light and transmittance values through network training [15]. Neural networks are usually used to learn the nonlinear mapping between foggy images and real images [16]. In recent years, this field has developed rapidly. Cai et al. [17] proposed a DehazeNet network based on Convolutional Neural Network (CNN) in their study for removing haze from images. The network estimates the atmospheric light value A and then substitutes A into the atmospheric scattering model to restore it to a fog free image. However, for certain specific scenarios where the estimation of the medium transmission image is inaccurate, defogging images may suffer from color distortion, excessive defogging, and other issues [36]. Moreover, the use of global atmospheric light values and block estimation methods results in a inference time of 5.09 seconds for a single 640×480 fog image on a general CPU [22]. Ren et al. [37] used a multi-scale convolutional neural network (MSCNN) to learn the mapping relationship between foggy images and media transport images. The network first trains a coarse medium transmission map of foggy images, then uses a fine medium transmission network to refine the coarse medium transmission map, and finally uses an atmospheric scattering model to restore clear images without fog. But it trains the input foggy image into a rough transmission matrix and then refines it, which inevitably leads to errors [38]. If the estimation error of the medium transmission diagram is large, the superposition of the errors of the two parameters

will deteriorate the dehazing effect. The AOD-Net designed by Li et al. [22] achieved the mapping from input foggy images to generated non foggy images. Using a large number of foggy and non foggy datasets in the same scene, an end-to-end network was trained from foggy images directly to non foggy images to avoid errors in estimating multiple atmospheric parameters. It does not separately estimate the transmission matrix and atmospheric light coefficient, but unifies the two parameters into one parameter through formula conversion, and finally generates clear images directly through lightweight CNN.

A. Mehra et al. [39] proposed ReViewNet, which is a fast, lightweight and powerful defogging system suitable for autonomous vehicle. The network uses components such as spatial feature pool, quadruple color prompt, multi view architecture and multi weighted loss to effectively de atomize images captured by autonomous vehicle cameras. Through detailed quantitative evaluation of five benchmark data sets, the effectiveness and progressiveness of the proposed model are analyzed.

H. Ullah [40] proposed a lightweight convolutional neural network called Light DehazeNet (LD-Net) with high computational efficiency for reconstructing hazy images. Unlike other deep learning methods that measure transmission maps and atmospheric light separately, LD-Net uses a transformed atmospheric scattering model to jointly estimate transmission maps and atmospheric light. In addition, combining color visibility restoration methods to avoid color distortion in dehazing images. Y. Jin et al. [41] designed a dehazing network called LFD-Net, which extracts, fuses, and weights multi-scale features through the network structure of a lightweight atmospheric scattering model. LFD-Net achieves strong interpretability by utilizing gating fusion modules and attention mechanisms to facilitate feature interaction between multi-level representations.

2.3. Hardware Implementation of Image Dehazing Algorithms

With the rapid development of integrated circuits, various digital chips such as microcontrollers, ASICs, FPGAs, DSPs, etc. have gradually emerged in the field of image processing. When using microcontrollers for image processing, due to limited storage space, it is impossible to cache and process large amounts of image data, and there is a problem of slow processing speed; ASIC needs to be customized according to specified requirements, which may result in long development cycles and poor flexibility; FPGA is widely used in the field of image processing due to its fast parallel processing speed, small size, low power consumption, and high flexibility. To address the issue of limited FPGA resources, Li et al. [42] optimized Gaussian kernels using address encoding and distributed algorithms, achieving parallel multi-scale convolution. This implementation itself does not require frame buffering, greatly saving the use of BlockRAM. UStukov et al. [43] used the dual port RAM (BRAM) inside the FPGA to store each frame and line of the image, with the depth of the BRAM equal to the width of the image. Jin. Woo. Park et al. [44] processed the data as a stream between each module from the input stage, using only the minimum row buffer without using frame buffers. Zhou [45] proposed a single image dehazing system based on FPGA. Firstly, the average statistical method is used to estimate the atmospheric light value. Then, the improved dark channel map is used to estimate the transmittance. Finally, the guided filtering algorithm is used to refine the transmittance map, and the fast median filtering method is used to adaptively process the image boundary. Guo et al. [46] proposed an adaptive sky segmentation image dehazing method and transplanted the algorithm to FPGA, but the dehazing effect of the algorithm is too dependent on the results of sky segmentation. Kosugel et al. [47] utilized the partial reconstruction capability of FPGA to reduce hardware resource consumption when studying the nearest point iteration algorithm. A low complexity estimation method for aerial lighting based on distributed sorting and scaling brightness, taking into account the consistency between consecutive images in the video stream, and reducing computational complexity through frame extraction. In FPGA, frame data is obtained through direct memory access (DMA) and enhanced by a dehazing processor. Zhang Haibin [48] achieved image dehazing by completing bilinear interpolation scaling module, dark channel map and transmittance

map calculation module on the PL side of ZYNQ platform, and auxiliary algorithm and system control functions on the PS side.

Overall, traditional image dehazing techniques can achieve good results in handling simple scenes, but their performance is often unsatisfactory when facing more complex haze environments. In addition, these methods rely on specific prior knowledge, which to some extent limits their adaptability and generalization ability to different types of haze conditions. In contrast, although deep learning based methods have shown great potential, there is still room for further optimization in terms of improving algorithm efficiency, reducing time complexity, and reducing resource consumption, whether based on neural networks or more advanced Transformer architectures.

3. METHOD

This section will introduce the following key parts in sequence: A) dehazing preprocessing and technical details, B) LDM-Net network architecture analysis, C) pixel attention mechanism design, D) FPGA hardware deployment design, E) explanation of the loss function used.

3.1. Dehazing Preprocessing and Technical Details

The Atmospheric Scattering Model (ASM) has become a classic benchmark in the field of image dehazing, but there is a problem of high time complexity in separately estimating $t(x)$ and A. In order to simplify this process and improve efficiency, this article has improved the ASM model, and its mathematical expression is shown in formula 2:

$$J(x) = \frac{I(x)-A}{t(x)} + A \tag{2}$$

Encapsulate $t(x)$ and A into an adaptive parameter $K(x)$ to estimate the haze free image $J(x)$, and the resulting mathematical expression is shown in formula 3:

$$J(x) = K(x) \times I(x) - K(x) + b \tag{3}$$

Overall, K here refers to a parameter used to evaluate and optimize the performance of image processing networks, typically associated with color channels such as RGB (red, green, and blue). At the end of the algorithm processing flow, a K value will be generated for each color channel of each input image, which will be used for further calculation or optimization. In addition, this article sets the b value to the most commonly used default value of 1.

3.2. Deep Learning Solutions

3.2.1. LDM-Net network architecture analysis.

Unlike most heavyweight image dehazing networks, LDM-Net, as a lightweight image dehazing network, enhances the interaction and learning between feature information by introducing concatenation layers and pixel attention mechanisms, improving the model's generalization ability and image dehazing performance without making the model too bulky. The LDM-Net network framework is shown in Fig. 1:

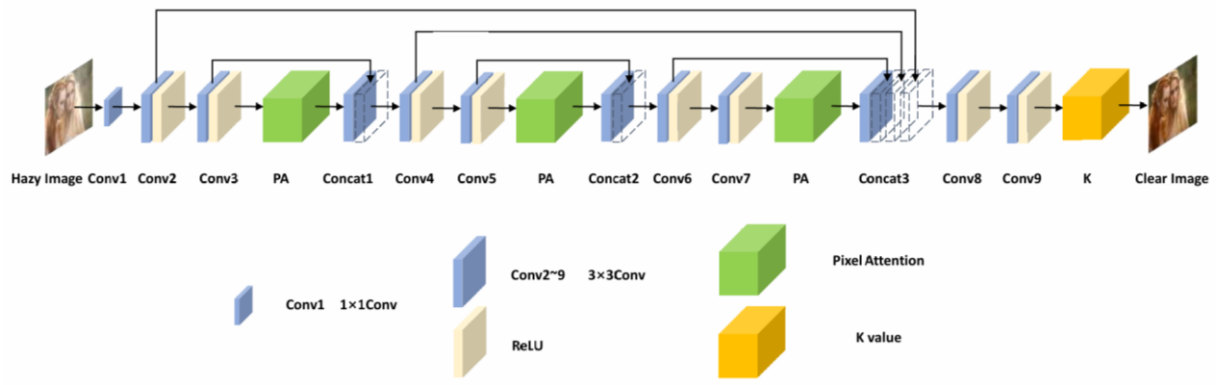

Figure 1. LDM-Net Framework Structure

In Convolutional Neural Networks (CNNs), the size of the convolution kernel affects the level and range of feature extraction. Large convolutional kernels capture global features, while small convolutional kernels excel at capturing local details. Although small convolutional kernels are widely used due to their performance and efficiency, lightweight models may face computational efficiency issues in deep networks. This article introduces a cascade layer to combine low-level and high-level features, optimizes the design of convolutional layers to simplify feature extraction, and adds an attention module to enhance feature interaction, improve feature fusion ability and model attention to important features, making the network more efficient and accurate in processing complex tasks.

This article uses three concatenated layers, Concat1, Concat2, and Concat3, for feature fusion to ensure that each layer's feature representation contributes to the final output. For example, the Concat1 layer integrates features from Conv1 and PA modules, retaining important information learned in the early stages. This hierarchical structure optimizes information retention and integration, significantly improving the dehazing effect of images. The specific network layer parameters are shown in Table 1.

Table 1. Network Framework

	Channel (In/Out)	Kernel Size	Padding	Stride
Conv1	2/6	1	0	1
Conv2	6/6	3	1	1
Conv3	6/6	3	1	1
Concat1	Conv1+PA			
Conv4	12/12	3	1	1
Conv5	12/12	3	1	1
Concat2	Conv4+PA			
Conv6	24/24	3	1	1
Conv7	24/24	3	1	1
Concat3	Conv2+Conv5+Conv6+PA			
Conv8	66/32	3	1	1
Conv9	32/3	3	1	1

Specifically, the input foggy image first passes through a convolutional layer Conv1, which has 2 input channels and 6 output channels. Here, a 1×1 convolutional kernel is used for feature extraction without any padding, with a stride set to 1 to maintain the spatial dimension unchanged. Next, the feature map enters the second convolutional layer Conv2, which has 6 input and output channels. Using a 3×3 convolution kernel can capture a wider range of contextual information while performing a 1-unit padding to maintain image size, with a stride of 1. Subsequently, the feature map is passed through the third convolutional layer Conv3, which has the same configuration as Conv2 and uses

3×3 convolution kernels. The input and output channels are both 6, and 1 unit of padding is applied with a stride of 1. After these three convolutional layers, a concatenation layer Concat1 is introduced. This special layer concatenates the output of Conv1 with the output processed by pixel attention mechanism in the depth direction. This operation helps to increase the expressive power of the model, enabling the network to learn more complex and abstract features. Afterwards, the feature map is passed through the fourth convolutional layer Conv4, which has 12 input and output channels. A 3×3 convolution kernel is used to fill 1 unit with a stride of 1. This step further enhances the abstraction level of the features. Next is the fifth convolutional layer Conv5, which has the same configuration as Conv4 and uses 3×3 convolution kernels. It has 12 input and output channels and is filled with 1 unit, with a stride of 1. Then it reaches the second concatenation layer Concat2, which concatenates the output of Conv4 with the output processed by pixel attention mechanism in the depth direction. This design enables the network to integrate features from different levels, enhancing the expressiveness of the model. Afterwards, the feature map is passed through the sixth convolutional layer Conv6, which has 24 input and output channels. A 3×3 convolution kernel is used to fill the feature map with 1 unit, with a stride of 1. Next is the seventh convolutional layer Conv7, which has the same configuration as Conv6 and uses 3×3 convolution kernels. It has 24 input and output channels and is filled with 1 unit of padding, with a stride of 1. Next is the third concatenation layer Concat3, which concatenates the outputs of Conv2, Conv5, and Conv6 with the output processed by pixel attention mechanism in the depth direction. This further enriches the feature representation capability of the model. Afterwards, the feature map is passed through the eighth convolutional layer Conv8, which has 66 input channels (obtained from the previous concatenation layer) and 32 output channels. A 3×3 convolution kernel is used to fill 1 unit with a stride of 1. This step effectively integrates the features of previous layers. Finally, the image passes through the ninth convolutional layer Conv9, which has 3 input and output channels. A 3×3 convolution kernel is used to fill the image with 1 unit of padding, with a stride of 1. This is the final convolutional operation of the entire network, which maps features to the final output space.

3.2.2. Pixel attention mechanism design.

In order to further improve the generalization ability and feature information interaction ability of the model, this paper is inspired by reference [49] and introduces pixel attention into the network. It is worth noting that this article has improved the introduced pixel attention mechanism by modifying the original method of multiplying pixel weights to adding weights. The purpose of doing so is to reduce the complexity of the model, improve its computational efficiency, and reduce its inference time. Specifically, the input feature vector is processed through a 1×1 convolutional layer and highlighted with a Softmax activation function to highlight important features while suppressing less relevant ones. The improved pixel attention mechanism is shown in Fig. 2:

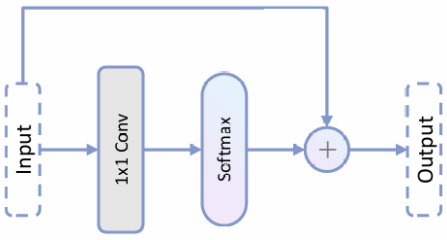

Figure 2. Pixel Attention

3.2.3. Loss Function.

Previous studies [19, 50-52] have shown that using loss functions such as L1 loss, L2 loss, and perceptual loss alone, or combining multiple loss functions, can demonstrate good performance in various tasks. However, this article found through multiple experiments that the combination of multiple loss functions is not suitable for the LDM Net model. On the contrary, the commonly used L1 loss function performs the best in this model. Specifically, the L1 loss function can effectively

optimize model parameters, thereby achieving better dehazing performance. The definition of L1 loss function is shown in formula 4:

$$L1(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left\|J^i - I^i\right\| \tag{4}$$

Among them, θ represents the network transmission parameter, J represents the true value after dehazing, and I represents the value of the input foggy image.

3.3. FPGA Hardware Deployment Design

3.3.1. Quantization.

In this study, we chose the ZYNQ-7000 series as the hardware deployment platform. Given the limited resources of the ZYNQ hardware platform and the challenges faced by FPGAs in processing floating-point operations, introducing model quantization techniques has become a necessary step. Currently, several mature quantization methods [53-56] have been widely applied in the optimization process of deep learning models. This study specifically utilized the quantization principle provided by TFLite [54] to perform quantization transformations on the key parameters of the model used. The core idea of this method is to simplify the calculation process by removing unnecessary product terms, and to enable the quantized model to run more efficiently in the processing logic (PL) part of ZYNQ. In addition, it also supports integer type data operations and displacement operations, which are crucial for improving overall system efficiency. Specifically, the conversion from floating-point numbers to fixed-point numbers can be described by two mathematical formulas: Formula 5 is used to define how to map the original floating-point values to a specific range; Formula 6 explains how to select the most suitable integer value within this new range to approximate the original value.

$$q_{float} = S\left(q_{integer} - z\right) \tag{5}$$

$$q_{integer} = round\left(\frac{q_{float}}{S} + z\right) \tag{6}$$

Among them, q_{float} represents floating-point numbers, $q_{integer}$ represents quantized integers, S represents the scale factor of the proportional relationship between real numbers and integers, and z represents zeros.

3.3.2. Model reconstruction.

In the LDM-Net network architecture, the convolutional layer, pooling layer, and activation function are the basic modules that constitute its core processing flow. In order to efficiently implement these key components and optimize their execution efficiency on hardware, using specific data structures in advanced synthesis (HLS) technology for cache management has become an important strategy. Specifically, through a carefully designed two-level integrated data structure, it can effectively support the large-scale data processing requirements in convolution operations, while ensuring that the computation process is both fast and energy-efficient. For the specific implementation of convolutional layers, a multi-level nested "for" loop structure is used to traverse every position of the input feature map and calculate the corresponding convolution kernel weights to generate the output feature map.

In this process, three key multidimensional arrays are first defined: the first represents the input feature map, the second is used to store intermediate or final output feature maps, and the third contains all the convolution kernel weight values involved in the current convolution operation. It is worth noting that each convolution kernel itself exists in the form of a three-dimensional array, which not only contains information on the size of the filter in the spatial dimension, but also specifically states that the number of input channels processed by the corresponding convolution kernel must be

consistent with the requirements specified in the overall network design. In order to further improve performance, several instructions were added when writing the above loop logic, aimed at guiding the compiler on how to better optimize the code. For example, a complete unroll strategy was adopted for the inner and outer control variables, which correspond to different channel indices in the loop body; For iterations involving the height and width directions of feature maps, the default state is maintained without any special processing. In addition, it was emphasized that additional extensions need to be made to the multidimensional arrays used along the channel dimension in order to fully utilize the parallel computing resources available in modern FPGA architectures. Similarly, when it comes to pooling layers, although there are slight differences in their functionality-mainly simplified versions of spatial down sampling rather than complex feature extraction - there are many commonalities in their implementation methods that are worth learning from. For example, a similar multi-layer loop structure can also be used to describe the entire pooling process, and the relevant directive settings can be adjusted according to the actual situation to achieve the best results. The convolution process is shown in Fig. 3:

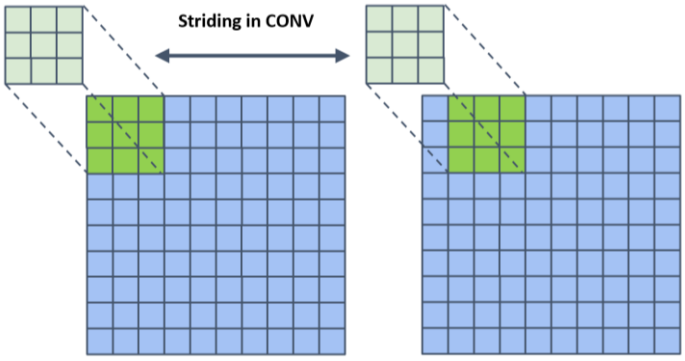

Figure 3. Convolution operation.

3.3.3. Hardware system.

The image is first input into the ZYNQ chip. The ZYNQ chip contains a hardware acceleration unit (PL) and a processing system (PS) internally. The image is transmitted from PL to VDMA module through the "Image to Stream" interface, and then returned to PS through the "Stream to Image" interface. The PS section includes a DDR3 controller, which is used to control the read and write operations of DDR3 memory, thereby storing and managing image data. Finally, the image data is output through HDMI. Specifically, foggy images are captured by a camera and converted into an image data stream, which is then read/written through a FIFO. The video stream storage architecture of VDMA captures video frames from external sources, streams them and buffers them in external memory DDR3, and configures the frame rate of VDMA frames with Dynamic Genlock Master to prevent image fragmentation. Display the processed image data stream through an HDMI external monitor. The operation process is shown in Fig. 4:

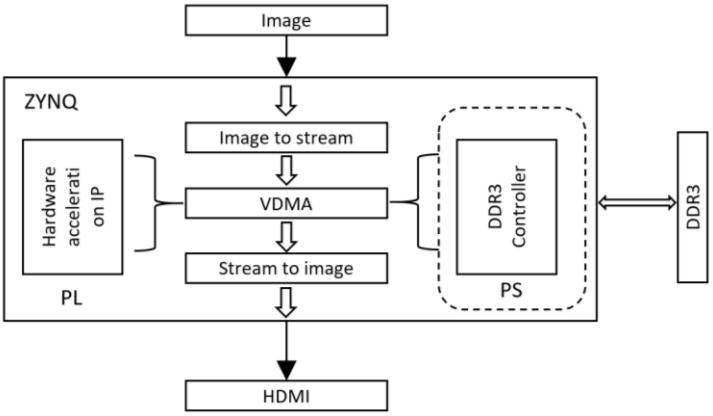

Figure 4. Hardware solution.

This article proposes an efficient method for utilizing the hardware resources of ZYNQ chips, which combines the advantages of distributed RAM and block RAM (BRAM). In addition to basic components such as Configurable Logic Blocks (CLBs) and switch interconnect matrices, the ZYNQ chip is also equipped with two key hardware resources: Block RAM and Hard Multiplicative Accumulator (DSP48E). The rational utilization of these resources is crucial for improving the overall computational performance of the system. With the support of hardware DSP48E, it can be ensured that two data items can be calculated within one clock cycle within a specific bit width range. In addition, block RAM can be used to cache intermediate calculation results, further optimizing processing speed. It is worth noting that these two resources are usually distributed in a logical array in columns, namely the PL part, and they are often close to each other. Therefore, the resources implemented by the array can be specified. For model parameter arrays with fewer parameters but higher performance requirements, distributed RAM, i.e. lookup tables (LUTs), is used for implementation. Due to the fact that distributed storage can be closer to the associated computational components, it can achieve better processing performance. In order to further improve data processing efficiency, this article adopts an array segmentation strategy. The main purpose of array partitioning is to increase the number of independent read-write ports, but more is not necessarily better. The specific number of partitions must be determined based on the actual data flow. Normally, there are three main forms of array partitioning: block partitioning, cyclic partitioning, and register partitioning. The block segmentation method first divides the data elements into several equal parts, and then fills each part in sequence; The cyclic segmentation method first divides the data into several equal parts, and then assigns one element to each block in turn, followed by a second round of allocation until all elements are allocated; The method of complete partitioning is relatively simple, which is achieved by allocating registers of a certain bit width to each array element. Among them, factor represents the number of segmentation blocks, and dim is the dimension of segmentation.

4. EXPERIMENT AND ANALYSIS

4.1. Experimental Setup

4.1.1. Datasets.

This section provides an overview of the datasets used in the training, validation, and testing phases of the LDM-Net model. During the training process, we used the OTS dataset, which contains 72135 outdoor blurry images composed of paired clean outdoor images and corresponding blurry images. These blurry images were generated based on 2061 real outdoor scene images from real-time weather. When training the LDM-Net model, we selected 75% of the images in the OTS dataset as the training set and used the remaining 25% of the images as the validation set. To comprehensively evaluate the performance of the model, we also introduced multiple test sets, including synthetic datasets and real-world datasets. For the evaluation of synthetic scenarios, this article selected the RESIDE [57] dataset to synthesize objective test sets (SOTS) and the Haze4K dataset [58] as the data sources for testing. The SOTS dataset contains 500 indoor and 500 outdoor test images, while the Haze4K dataset contains 4000 synthetic images. For the evaluation of real-world scenarios, this article selected a mixed subjective test set (HSTS) and real outdoor image datasets with and without fog to further validate the performance of the model in practical applications.

4.1.2. Training parameters.

This study utilized the PyTorch deep learning framework and implemented the proposed LDM Net model on a computing platform equipped with NVIDIA GeForce RTX 3060 Laptop GPU. In the optimization process, we chose Adam [59] algorithm as the optimizer and trained it for 40 epochs. Correspondingly, set the parameters $\beta 1$, $\beta 2$, and ε to 0.9, 0.999, and 1e-7, respectively. In addition, the initial learning rate of the model is set to 1e-4, the batch processing size is 16, and a cosine

annealing strategy is adopted [60] to adjust the learning rate through a cosine function. The mathematical expression for cosine annealing is shown in formula 7.

$$lr(t) = \frac{1}{2}\left(1 + cos\left(\frac{t\pi}{T}\right)\right) \times lr_{init} \quad (7)$$

Among them, t represents the current number of training steps, T represents the total number of training steps, and lr_{init} represents the initial learning rate. In addition, during the training process of the model, this article also performs data augmentation operations by randomly cropping small blocks of 256×256 pixels from the original image, rotating the image by 90°, 180°, or 270°, and flipping it vertically or horizontally.

4.1.3. Evaluation metrics.

In the field of image dehazing, peak signal-to-noise ratio (PSNR) [61] and structural similarity (SSIM) [62] are important indicators for evaluating image quality. This article uses them to measure the effectiveness and performance of image dehazing processing. In addition, time complexity (s) and the number of model parameters $(\times Param)$ were introduced to evaluate the overall performance of the model.

4.2. Experimental Evaluation

This section will compare LDM-Net with other advanced image dehazing methods, starting from parameters such as SSIM, PSNR, model parameters, and inference time, and combining visual observation to objectively evaluate model performance. Among them, the best performing results are presented in bold black.

4.2.1. Deep learning model performance.

As shown in Table 2, the performance of the proposed method and state-of-the-art technology (SOTA) on different datasets was compared. The results showed that among various image dehazing methods, the performance of manual prior based DCP [1] was not satisfactory. On all datasets, the PSNR value exceeded 20, and the SSIM value did not exceed 0.82. These results reveal the hypothetical limitations of the DCP method in handling different environmental conditions. Although AOD-Net [22], a deep learning based image dehazing method, performs well with PSNR values exceeding 20 on the Indoor SOTS dataset, it does not show outstanding performance on other datasets. Other deep learning based methods [40, 63, 64] also have similar problems, with excellent PSNR and SSIM values on one or two datasets, but there are still some datasets that perform poorly. In contrast, the method proposed in this article outperforms other dehazing methods in terms of PSNR and SSIM metrics. Specifically, LDM-Net has PSNR values exceeding 25 and SSIM values exceeding 0.96 on multiple datasets.

Table 2. The average values of PSNR and SSIM on SOTS dataset, Haze4K dataset, and HSTS dataset

Method	Indoor SOTS		Outdoor SOTS		Haze4K		HSTS	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DCP	18.88	0.8129	17.68	0.7956	18.01	0.8152	17.88	0.8103
AOD-Net	20.05	0.8477	19.28	0.8582	18.05	0.8335	18.01	0.8032
FFA-Net	23.66	0.8702	20.56	0.9103	18.92	0.8434	18.99	0.8233
LD-Net	22.12	0.8532	19.86	0.9032	19.56	0.8843	22.53	0.8941
CDC-Net	22.78	0.9122	20.33	0.9231	18.55	0.8285	28.69	0.8743
LDM-Net(Ours)	24.52	0.9433	26.78	0.9523	21.63	0.9229	23.12	0.9346

In the field of image dehazing, relying solely on quantitative analysis is not sufficient to comprehensively and objectively evaluate the performance of dehazing algorithms, and visual inspection (qualitative research) must be supplemented. In theory, the higher the peak signal-to-noise ratio (PSNR) value and the closer the structural similarity index (SSIM) value is to 1, the better the dehazing effect of the image. However, in practice, there may be situations where PSNR and SSIM values perform well, but visually the image quality after defogging is not ideal. Therefore, this study introduces a qualitative evaluation method aimed at comprehensively and reasonably evaluating the dehazing performance of images.

As shown in Fig. 5, the DCP dehazing method based on prior knowledge exhibits significant shortcomings in dehazing indoor scenes, failing to achieve thorough dehazing on a regional basis; In outdoor applications, the effect is even worse, as the dehazed image shows shadows and distortions in blank areas. Some deep learning based image dehazing techniques also have similar problems, especially AOD-Net, which performs the worst in dehazing among deep learning methods. Both indoor and outdoor image dehazing did not achieve the expected results. In contrast, the method proposed in this article demonstrates superior performance in both indoor and outdoor image dehazing. From a visual perspective, the dehazing results of indoor images are close to natural real images, and the dehazing of outdoor images has basically reached the level of real images.

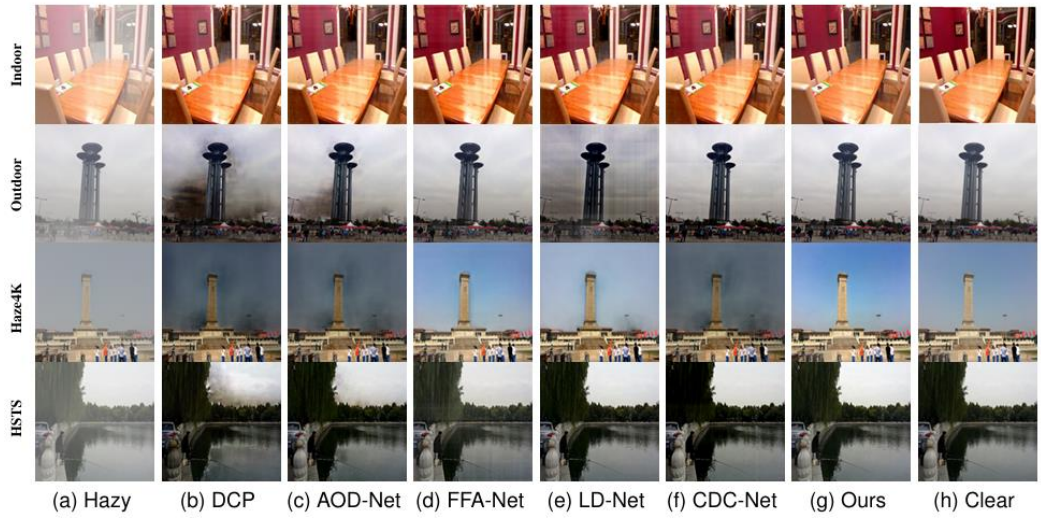

Figure 5. Qualitative comparison on SOTS dataset, OTS dataset, and Haze4K dataset.

4.2.2. Inference time and Model parameters.

In the evaluation of deep learning models, inference time and model parameter quantity are two key performance indicators. This study used the Haze4K dataset to objectively evaluate the computational time complexity of each model by comparing the average inference time of different deep learning image dehazing methods. The experimental environment was configured with a 3.20 GHz AMD Ryzen 7 6800H Radeon with Radeon Graphics CPU, 16 GB RAM, and NVIDIA GeForce RTX 3060 Laptop GPU, and conducted in PyTorch 2.0.1 environment.

As shown in Table 3, the FFA-Net [63] image dehazing method performs poorly, with an average processing time of 1.24 seconds for a single haze image with a resolution of 400×400. In contrast, the LDM-Net image dehazing network proposed in this article performs superior and has the shortest image dehazing execution time among all compared methods. Specifically, LDM-Net takes an average of only 0.27 seconds to process a single fog image with a resolution of 400×400.

Table 3. Average Execution Time

Method	AOD-Net	FFA-Net	LD-Net	CDC-Net	LDM-Net(Ours)
Time/s	0.65	1.24	0.41	0.37	0.27

As shown in Table 4, although AOD-Net has the smallest number of model parameters, combined with the qualitative and quantitative evaluation analysis mentioned above, AOD-Net's performance in image dehazing is not ideal. This indicates that AOD-Net sacrifices the performance of image dehazing by reducing model parameters. It is worth noting that the method proposed in this article has an advantage over most other methods in terms of model parameters, that is, it is more lightweight. Specifically, the model parameters of LDM-Net are close to AOD-Net, and it performs well in terms of dehazing performance indicators. This indicates that LDM-Net can achieve excellent image dehazing effects while maintaining low time complexity and fewer model parameters.

Table 4. Model Parameter

Method	AOD-Net	FFA-Net	LD-Net	CDC-Net	LDM-Net(Ours)
Param (×M)	0.002	4.406	3.008	0.098	0.027

4.2.3. FPGA hardware performance.

This study selected a traditional DCP dehazing method based on prior knowledge and three lightweight image dehazing models based on deep learning for deployment on FPGA for comparison. As shown in Table 5, the DCP image dehazing method still performs poorly, with unsatisfactory PSNR and SSIM values on multiple datasets. Although the deep learning based image dehazing model is affected by quantization and its performance has declined, overall it is still superior to traditional image dehazing methods. It is worth noting that the LDM-Net image dehazing network proposed in this article performs the best on various datasets, with the best PSNR and SSIM values. Specifically, when the LDM-Net model runs on the FPGA framework, the average PSNR values exceed 20 and SSIM values exceed 0.9. This result indicates that LDM-Net can still maintain excellent dehazing performance in hardware acceleration environments.

Table 5. The average values of PSNR and SSIM on SOTS dataset, Haze4K dataset, and HSTS dataset

Method	Indoor SOTS		Outdoor SOTS		Haze4K		HSTS	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DCP	18.62	0.8012	18.68	0.8132	17.91	0.8052	17.86	0.8023
AOD-Net	18.15	0.8357	18.42	0.8502	18.15	0.8232	17.91	0.8132
LD-Net	20.32	0.8412	18.81	0.8812	18.52	0.8543	21.23	0.8847
LDM-Net(Ours)	22.32	0.9341	24.84	0.9413	20.83	0.9021	22.09	0.9046

As shown in Fig. 6, it can be clearly observed through visual observation that the performance of the DCP method is still unsatisfactory, with large areas of black shadows and distortion in the dehazed image. In addition, the performance of the selected deep learning based image dehazing algorithm also showed a certain degree of decline after 8-bit quantization and deployment on the FPGA platform. Especially in the AOD-Net model, serious black shadow and distortion problems also appeared in the processed results, especially when facing large blank areas, this phenomenon is more significant. In contrast, the hardware acceleration framework proposed in this article demonstrates stronger robustness to the above challenges. Even after quantization and hardware implementation, the final generated dehazing image can still retain the details of the original scene well, approaching the ideal clear state. This indicates that the method proposed in this study effectively reduces the negative impact caused by hardware limitations while maintaining a high defogging effect.

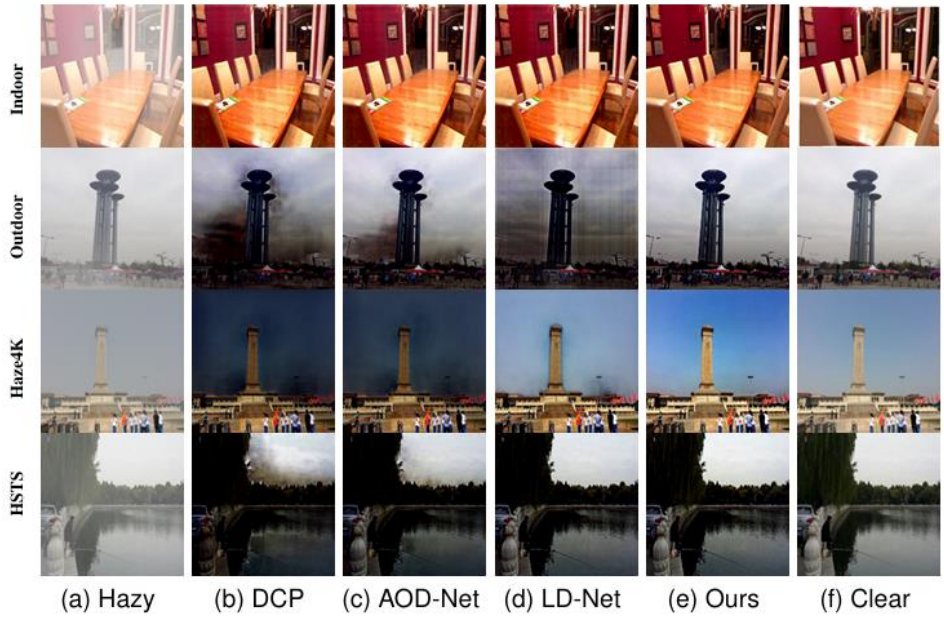

Figure 6. Qualitative comparison on SOTS dataset, OTS dataset, and Haze4K dataset.

4.2.4. Inference time and Resource consumption.

After successfully deploying the image dehazing algorithm on the FPGA platform, the introduction of ping-pong flow mechanism significantly optimized the processing flow, thereby greatly improving the average inference speed of dehazing operations on foggy images. According to the data in Table 6, three deep learning based image dehazing schemes, including the method proposed in this study, have achieved varying degrees of performance improvement in hardware acceleration environments. Taking a single foggy image with a size of 400×400 pixels as an example, the AOD-Net model takes an average of approximately 0.631 seconds to complete a complete dehazing process; And LD-Net [40] can achieve the same goal in a shorter time (i.e. 0.082 seconds); It is worth noting that with the design method proposed in this article, the task can be completed in only about 0.021 seconds. From this, it can be seen that compared to the other two deep learning driven image dehazing techniques, when applied to dedicated hardware devices such as FPGA, the proposed solution in this study exhibits superior time efficiency. It not only significantly shortens the processing cycle of a single sample, but also means that it has stronger real-time performance and practicality in large-scale application scenarios. This achievement fully demonstrates the effectiveness of our optimization work for specific hardware architectures, and lays a solid foundation for further exploration of high-performance and low latency visual computing solutions in the future.

Table 6. Average Execution Time

Method	AOD-Net	LD-Net	LDM-Net(Ours)
Time/s	0.631	0.082	0.021

When deploying deep learning models on FPGA platforms, in addition to focusing on processing speed, it is also necessary to fully consider the important indicator of hardware resource consumption. As shown in Table 7, taking the LDM Net model as an example, after completing the migration to the FPGA end and performing system level synthesis, its occupancy of various key hardware resources is as follows: the actual usage of BRAM18K memory is 120 units, accounting for 11.5% of the total available quantity; The DSP48E digital signal processing unit has been called 590 times, accounting for 65.3% of the overall resource pool; The Flip Flop component activated a total of 130676 instances, resulting in a resource utilization rate of 37.7% for this class; In terms of lookup tables (LUTs), as many as 130049 specific applications were recorded, increasing the proportion of resource allocation for this item to 75.7%. These data indicate that although specific optimization

methods can alleviate the challenges caused by model complexity to some extent, how to reasonably control resource overhead while ensuring high efficiency remains a major challenge in current research. In future work, more efficient algorithm design and architecture adjustment strategies should be further explored in order to achieve the optimal balance between performance and cost.

Table 7. Hardware resource consumption

Hardware resources	Actual consumption	Resources consumption rate/%
BRAM_18K	120	11.5
DSE48E	590	65.3
FF	130676	37.7
LUT	130049	75.7

5. DISCUSSION

Through in-depth analysis of qualitative and quantitative evaluation experiments, this article draws the following comprehensive conclusions. Firstly, the LDM-Net dehazing model proposed in this article demonstrates its unique advantages in objective evaluation indicators. The performance in PSNR and SSIM metrics is very impressive, surpassing current advanced dehazing methods. This result not only demonstrates the outstanding ability of LDM-Net in maintaining image structural information, but also reflects its enormous potential in improving image quality after dehazing. Secondly, from a visual observation perspective, LDM-Net also demonstrated its excellent dehazing effect. On most of the test datasets, the images processed by LDM-Net are clear, rich in details, effectively remove fog, and have a better visual experience than other comparison methods. This excellent visual effect further validates the reliability and effectiveness of LDM-Net in practical applications. It is worth mentioning that LDM-Net also has significant advantages in model parameters and execution time. Compared to other high-performance dehazing models, LDM-Net has fewer model parameters, which means it requires fewer resources for deployment and is more suitable for running in resource constrained environments. At the same time, LDM-Net has a shorter average time for single image dehazing, which greatly improves its efficiency in practical application scenarios. In addition, this article also implemented hardware deployment of the LDM-Net model. Considering the difficulty of FPGA in computing floating-point numbers, this paper quantifies the LDM-Net model. Finally, based on the dehazing performance of the model running on the FPGA framework, it also outperforms other deep learning models running on the FPGA framework.

Overall, this study successfully achieved efficient and high-quality image dehazing by proposing the LDM-Net model. The experimental results indicate that LDM-Net performs well in multiple key performance indicators, including objective evaluation metrics such as PSNR and SSIM, as well as subjective visual experience. In addition, LDM-Net also has significant advantages in model parameters and execution time, making it more suitable for application in resource constrained environments. Future research can further optimize the LDM-Net model to adapt to image dehazing tasks in more complex scenarios, and explore its deployment possibilities on more hardware platforms.

REFERENCES

[1] K. He, J. Sun, and X. Tang, “Single Image Haze Removal Using Dark Channel Prior,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 12, pp. 2341–2353, Dec. 2011, doi: 10.1109/TPAMI.2010.168.

[2] B. B. Upadhyay and K. Sarawadekar, “VLSI Design of Saturation-Based Image Dehazing Algorithm,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 31, no. 7, pp. 959–968, Jul. 2023, doi: 10.1109/TVLSI.2023.3272018.

[3] Y.-K. Wang and C.-T. Fan, “Single Image Defogging by Multiscale Depth Fusion,” IEEE Transactions on Image Processing, vol. 23, no. 11, pp. 4826–4837, Nov. 2014, doi: 10.1109/TIP.2014.2358076.

[4] Q. Zhu, J. Mai, and L. Shao, “A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior,” IEEE Transactions on Image Processing, vol. 24, no. 11, pp. 3522–3533, Nov. 2015, doi: 10.1109/TIP.2015.2446191.

[5] Y. Xu, J. Wen, L. Fei, and Z. Zhang, “Review of Video and Image Defogging Algorithms and Related Studies on Image Restoration and Enhancement,” IEEE Access, vol. 4, pp. 165–188, 2016, doi: 10.1109/ACCESS.2015.2511558.

[6] Y. Li et al., “Underwater Image High Definition Display Using the Multilayer Perceptron and Color Feature-Based SRCNN,” IEEE Access, vol. 7, pp. 83721–83728, 2019, doi: 10.1109/ACCESS.2019.2925209.

[7] J. Zhou, D. Zhang, P. Zou, W. Zhang, and W. Zhang, “Retinex-Based Laplacian Pyramid Method for Image Defogging,” IEEE Access, vol. 7, pp. 122459–122472, 2019, doi: 10.1109/ACCESS.2019.2934981.

[8] R. Kumar, B. K. Kaushik, and R. Balasubramanian, “Multispectral Transmission Map Fusion Method and Architecture for Image Dehazing,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 11, pp. 2693–2697, Nov. 2019, doi: 10.1109/TVLSI.2019.2932033.

[9] W. Wang, X. Wu, X. Yuan, and Z. Gao, “An Experiment-Based Review of Low-Light Image Enhancement Methods,” IEEE Access, vol. 8, pp. 87884–87917, 2020, doi: 10.1109/ACCESS.2020.2992749.

[10] Ren Yanfei, "Application of Histogram Equalization in Image Processing," Science and Technology Information, vol. 4, pp. 37-38, 2007.

[11] Dong Jingwei, Zhao Chunli, and Hai Bo, "Research on Image dehazing Algorithm Integrating Homomorphic Filtering and Wavelet Transform," Journal of Harbin Institute of Technology, no. 1, pp. 66–70, 2019, doi: 10.15938/j.jhust.2019.01.011.

[12] E. H. Land and J. J. McCann, “Lightness and Retinex Theory,” J. Opt. Soc. Am., vol. 61, no. 1, p. 1, Jan. 1971, doi: 10.1364/JOSA.61.000001.

[13] R. Fattal, “Single image dehazing,” ACM Trans. Graph., vol. 27, no. 3, pp. 1–9, Aug. 2008, doi: 10.1145/1360612.1360671.

[14] J.-P. Tarel and N. Hautière, “Fast visibility restoration from a single color or gray level image,” in 2009 IEEE 12th International Conference on Computer Vision, Sep. 2009, pp. 2201–2208. doi: 10.1109/ICCV.2009.5459251.

[15] Xu Yunqian, "Research on Image dehazing Algorithm and Its FPGA Implementation," Master's Thesis, China University of Mining and Technology, 2024 doi: 10.27623/d.cnki.gzkyu.2023.002561.

[16] Shao Shuai, "Research and System Implementation of Deep Learning Methods for Remote Sensing Image dehazing," Master's Thesis, Xi'an University of Technology, 2024 doi: 10.27398/d.cnki.gxalu.2023.000381.

[17] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao, “DehazeNet: An End-to-End System for Single Image Haze Removal,” IEEE Transactions on Image Processing, vol. 25, no. 11, pp. 5187–5198, Nov. 2016, doi: 10.1109/TIP.2016.2598681.

[18] C. Li, J. Guo, F. Porikli, H. Fu, and Y. Pang, “A Cascaded Convolutional Neural Network for Single Image Dehazing,” IEEE Access, vol. 6, pp. 24877–24887, 2018, doi: 10.1109/ACCESS.2018.2818882.

[19] J. Wang, Y. Xu, and W. Chen, “Multi-Patch and Feature Fusion Network for Single Image Dehazing,” in 2020 International Conference on Intelligent Computing, Automation and Systems (ICICAS), Dec. 2020, pp. 282–285. doi: 10.1109/ICICAS51530.2020.00064.

[20] Z. Chen, Y. Wang, and Y. Zou, “Inverse Atmoshperic Scattering Modeling with Convolutional Neural Networks for Single Image Dehazing,” in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Apr. 2018, pp. 2626–2630. doi: 10.1109/ICASSP.2018.8462078.

[21] G. Tang, L. Zhao, R. Jiang, and X. Zhang, “Single Image Dehazing via Lightweight Multi-scale Networks,” in 2019 IEEE International Conference on Big Data (Big Data), Dec. 2019, pp. 5062–5069. doi: 10.1109/BigData47090.2019.9006075.

[22] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, “AOD-Net: All-in-One Dehazing Network,” in 2017 IEEE International Conference on Computer Vision (ICCV), Oct. 2017, pp. 4780–4788. doi: 10.1109/ICCV.2017.511.

[23] Y. Zhang, F. Wang, and D. Yin, “Pixel Transformer for Synthetic-to-Real Single Image Dehazing,” in 2023 8th International Conference on Communication, Image and Signal Processing (CCISP), Nov. 2023, pp. 250–254. doi: 10.1109/CCISP59915.2023.10355800.

[24] Y. Song, Z. He, H. Qian, and X. Du, “Vision Transformers for Single Image Dehazing,” IEEE Transactions on Image Processing, vol. 32, pp. 1927–1941, 2023, doi: 10.1109/TIP.2023.3256763.

[25] Y. Xing, Z. Yan, T. Xiao, and X. Zhang, “TransID: Image Dehazing Based on TransUNet,” in 2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), Aug. 2022, pp. 508–512. doi: 10.1109/PRAI55851.2022.9904061.

[26] S. Medasani, B. Shireesha, B. Hemanth Reddy, C. C. Teja, and E. V. Sainath Chowdary, “Enhanced Single Remote Sensing Image Dehazing via Vision Transformer with Silencing Map Transmission and advanced Image Processing Techniques,” in 2024 International Conference on Intelligent Systems for Cybersecurity (ISCS), May 2024, pp. 1–6. doi: 10.1109/ISCS61804.2024.10581366.

[27] P. L. Suárez, D. Carpio, A. D. Sappa, and H. O. Velesaca, “Transformer based Image Dehazing,” in 2022 16th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Oct. 2022, pp. 148–154. doi: 10.1109/SITIS57111.2022.00037.

[28] Qian Wen, "Research on Image dehazing Algorithm Based on Deep Learning," Master's Thesis, Nanjing University of Posts and Telecommunications, 2021 doi: 10.27251/d.cnki.gnjdc.2020.000346.

[29] E. H. Land, “The Retinex Theory of Color Vision,” Sci Am, vol. 237, no. 6, pp. 108–128, Dec. 1977, doi: 10.1038/scientificamerican1277-108.

[30] D. H. Brainard and B. A. Wandell, “Analysis of the retinex theory of color vision,” J Opt Soc Am A, vol. 3, no. 10, pp. 1651–1661, Oct. 1986, doi: 10.1364/josaa.3.001651.

[31] A. Baskar and T. G. Kumar, “Automatic Face Enhancement Technique using Sigmoid Normalization based on Single Scale Retinex (SSR) Algorithm,” International Journal of Advanced Intelligence Paradigms, vol. 1, no. 1, 637450560000000000, doi: 10.1504/ijaip.2021.10030669.

[32] D. J. Jobson, Z. Rahman, and G. A. Woodell, “A multiscale retinex for bridging the gap between color images and the human observation of scenes,” IEEE Transactions on Image Processing, vol. 6, no. 7, pp. 965–976, Jul. 1997, doi: 10.1109/83.597272.

[33] S. Parthasarathy and P. Sankaran, “Fusion based Multi Scale RETINEX with Color Restoration for image enhancement,” in 2012 International Conference on Computer Communication and Informatics, Jan. 2012, pp. 1–7. doi: 10.1109/ICCCI.2012.6158793.

[34] Li Yadong, "Research on Key Technologies of Unmanned Aerial Vehicle Forest Resource Category II Survey Photogrammetry System," Ph.D. Dissertation, Beijing Forestry University, 2022 doi: 10.26949/d.cnki.gblyu.2020.001486.

[35] R. T. Tan, N. Pettersson, and L. Petersson, “Visibility Enhancement for Roads with Foggy or Hazy Scenes,” in 2007 IEEE Intelligent Vehicles Symposium, Jun. 2007, pp. 19–24. doi: 10.1109/IVS.2007.4290085.

[36] Chen Rui, "Research on Image dehazing Algorithm Based on Convolutional Neural Network and FPGA Implementation," Master's Thesis, Xi'an University of Technology, 2021 doi: 10.27398/d.cnki.gxalu.2020.001373.

[37] Y. Park and T.-H. Kim, “Fast Execution Schemes for Dark-Channel-Prior-Based Outdoor Video Dehazing,” IEEE Access, vol. 6, pp. 10003–10014, 2018, doi: 10.1109/ACCESS.2018.2806378.

[38] Li Xinchao, "Research on Image Defogging Algorithm Based on AOD-NET," Master's Thesis, Jilin University, 2024 doi: 10.27162/d.cnki.gjlin.2023.005242.

[39] A. Mehra, M. Mandal, P. Narang, and V. Chamola, “ReViewNet: A Fast and Resource Optimized Network for Enabling Safe Autonomous Driving in Hazy Weather Conditions,” IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 7, pp. 4256–4266, Jul. 2021, doi: 10.1109/TITS.2020.3013099.

[40] H. Ullah et al., “Light-DehazeNet: A Novel Lightweight CNN Architecture for Single Image Dehazing,” IEEE Transactions on Image Processing, vol. 30, pp. 8968–8982, 2021, doi: 10.1109/TIP.2021.3116790.

[41] Y. Jin, J. Chen, F. Tian, and K. Hu, “LFD-Net: Lightweight Feature-Interaction Dehazing Network for Real-Time Remote Sensing Tasks,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 16, pp. 9139–9153, 2023, doi: 10.1109/JSTARS.2023.3312515.

[42] Y. Li, H. Zhang, Y. You, and M. Sun, “A multi-scale retinex implementation on FPGA for an outdoor application,” in 2011 4th International Congress on Image and Signal Processing, Oct. 2011, pp. 1788–1792. doi: 10.1109/CISP.2011.6100606.

[43] D. I. Ustukov, Y. R. Muratov, and V. N. Lantsov, “Modification of retinex algorithm and its stream implementation on FPGA,” in 2017 6th Mediterranean Conference on Embedded Computing (MECO), Jun. 2017, pp. 1–4. doi: 10.1109/MECO.2017.7977246.

[44] J. W. Park et al., “A Low-Cost and High-Throughput FPGA Implementation of the Retinex Algorithm for Real-Time Video Enhancement,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 1, pp. 101–114, Jan. 2020, doi: 10.1109/TVLSI.2019.2936260.

[45] Z. Zhou and Z. Pan, “Real-time defogging hardware accelerator based on improved dark channel prior and adaptive guided filtering,” Journal of Electronic Imaging (JEI), 2022, Accessed: Nov. 06, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Real-time-defogging-hardware-accelerator-based-on-Zhou-Pan/ad31b6f24cab661b11363d34837337c98d62b3d3

[46] F. Guo, J. Qiu, and J. Tang, “Single Image Dehazing Using Adaptive Sky Segmentation,” IEEJ Transactions Elec Engng, vol. 16, no. 9, pp. 1209–1220, Sep. 2021, doi: 10.1002/tee.23419.

[47] F. Yu, Z. Zhang, H. Shen, Y. Huang, S. Cai, and S. Du, “FPGA implementation and image encryption application of a new PRNG based on a memristive Hopfield neural network with a special activation gradient,” Chinese Phys. B, vol. 31, no. 2, p. 020505, Jan. 2022, doi: 10.1088/1674-1056/ac3cb2.

[48] Zhang Haibin, "Design of Real time Video Image dehazing System Based on Zynq," Master's Thesis, Shanghai Normal University, 2018 Accessed: Nov. 06, 2024. [Online]. Available: https://kns.cnki.net/KCMS/detail/detail.aspx?dbcode=CMFD&dbname=CMFD201802&filename=1018216452.nh

[49] H. Zhao, X. Kong, J. He, Y. Qiao, and C. Dong, Efficient Image Super-Resolution Using Pixel Attention. 2020.

[50] Y. Guo, J. Chen, X. Ren, A. Wang, and W. Wang, "Joint Raindrop and Haze Removal From a Single Image," IEEE Transactions on Image Processing, vol. 29, pp. 9508–9519, 2020, doi: 10.1109/TIP.2020.3029438.

[51] M. S. Rad, B. Bozorgtabar, U.-V. Marti, M. Basler, H. K. Ekenel, and J.-P. Thiran, "SROBB: Targeted Perceptual Loss for Single Image Super-Resolution," in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Oct. 2019, pp. 2710–2719. doi: 10.1109/ICCV.2019.00280.

[52] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss Functions for Image Restoration With Neural Networks," IEEE Transactions on Computational Imaging, vol. 3, no. 1, pp. 47–57, Mar. 2017, doi: 10.1109/TCI.2016.2644865.

[53] N. Chandrinos, M. Amasialidis, M. Kirtas, K. Tsampazis, N. Passalis, and A. Tefas, "Quantization-Aware Training for Multi-Agent Reinforcement Learning," in 2024 32nd European Signal Processing Conference (EUSIPCO), Aug. 2024, pp. 1891–1895. Accessed: Nov. 09, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10715228

[54] B. Jacob et al., "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference," Dec. 15, 2017, arXiv: arXiv:1712.05877. doi: 10.48550/arXiv.1712.05877.

[55] Y. Zhang et al., "A Deep Learning Inference Scheme Based on Pipelined Matrix Multiplication Acceleration Design and Non-uniform Quantization," in 2021 IEEE 7th International Conference on Cloud Computing and Intelligent Systems (CCIS), Nov. 2021, pp. 281–285. doi: 10.1109/CCIS53392.2021.9754668.

[56] J. Jung, J. Kim, Y. Kim, and C. Kim, "Reinforcement Learning-Based Layer-Wise Quantization For Lightweight Deep Neural Networks," in 2020 IEEE International Conference on Image Processing (ICIP), Oct. 2020, pp. 3070–3074. doi: 10.1109/ICIP40778.2020.9191267.

[57] B. Li et al., "Benchmarking Single-Image Dehazing and Beyond," IEEE Transactions on Image Processing, vol. 28, no. 1, pp. 492–505, Jan. 2019, doi: 10.1109/TIP.2018.2867951.

[58] Y. Liu et al., "From Synthetic to Real: Image Dehazing Collaborating with Unlabeled Real Data," in Proceedings of the 29th ACM International Conference on Multimedia, in MM '21. New York, NY, USA: Association for Computing Machinery, Oct. 2021, pp. 50–58. doi: 10.1145/3474085.3475331.

[59] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," CoRR, Dec. 2014, Accessed: Aug. 15, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Adam%3A-A-Method-for-Stochastic-Optimization-Kingma-Ba/a6cb366736791bcccc5c8639de5a8f9636bf87e8

[60] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of Tricks for Image Classification with Convolutional Neural Networks," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2019, pp. 558–567. doi: 10.1109/CVPR.2019.00065.

[61] M. Martini, "A Simple Relationship Between SSIM and PSNR for DCT-Based Compressed Images and Video: SSIM as Content-Aware PSNR," in 2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP), Sep. 2023, pp. 1–5. doi: 10.1109/MMSP59012.2023.10337706.

[62] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.

[63] X. Qin, Z. Wang, Y. Bai, X. Xie, and H. Jia, "FFA-Net: Feature Fusion Attention Network for Single Image Dehazing," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 07, Art. no. 07, Apr. 2020, doi: 10.1609/aaai.v34i07.6865.

[64] Y. Zhong, J. Liu, X. Huang, J. Liu, Y. Fan, and M. Wu, "CDCNet: A Fast and Lightweight Dehazing Network with Color Distortion Correction," in ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Apr. 2024, pp. 3020–3024. doi: 10.1109/ICASSP48485.2024.10447111.