

Article

A Symmetric Multiprocessor System-on-a-Chip-Based Solution for Real-Time Image Dehazing

Dat Ngo ¹  and Bongsoon Kang ^{2,*} 

¹ Department of Computer Engineering, Korea National University of Transportation, Chungju 27469, Republic of Korea; datngo@ut.ac.kr

² Department of Electronic Engineering, Dong-A University, Busan 49315, Republic of Korea

* Correspondence: bongsoon@dau.ac.kr; Tel.: +82-51-200-7703

Abstract: The acquisition of digital images is susceptible to haze, and images captured under such adverse conditions may impact high-level applications designed for clean input data. Image dehazing emerges as a practical solution to this problem, as it can be employed to pre-process images immediately after acquisition. This paper presents a concise review of impactful algorithms, including those based on deep learning models, to identify the existing gap in real-time processing capabilities. Subsequently, a real-time dehazing system on a multiprocessor system-on-a-chip (MPSoC) platform is introduced to bridge this gap. The proposed system balances the trade-off between dehazing performance and computational complexity; hence, the name “Symmetric” is coined. Additionally, the entire system is implemented in programmable logic and wrapped by an interface circuit supporting double-buffering, rendering it highly suitable for seamless integration into existing camera systems. Implementation results on a Zynq UltraScale+ MPSoC ZCU106 Evaluation Kit demonstrate a maximum operating frequency of 356.51 MHz, equivalent to a maximum processing speed of 40.27 frames per second for DCI 4K resolution.

Keywords: image dehazing; deep learning; real-time processing; MPSoC



Citation: Ngo, D.; Kang, B. A

Symmetric Multiprocessor System-on-a-Chip-Based Solution for Real-Time Image Dehazing. *Symmetry* **2024**, *16*, 653. <https://doi.org/10.3390/sym16060653>

Academic Editors: Aleksandar Djordjevic and Miladin Stefanovic

Received: 9 April 2024

Revised: 21 May 2024

Accepted: 22 May 2024

Published: 25 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Image dehazing (also known as image defogging or visibility restoration) is a long-standing problem in computer vision due to its ill-posed nature. One of the earliest attempts at addressing this challenging problem dates back to the work of Vincent [1] in 1972. Over the years, a myriad of relevant studies have emerged, ranging from heuristic approaches (such as enhancement-based [2,3] and prior-based [4,5] methods) to data-driven techniques (such as deep learning methods [6–9]). As a result, the research field has now matured, with a strong focus on practical and application-oriented solutions, where dehazing algorithms are required to be computationally efficient for broad deployment. An illustrative example of this trend can be found in Adobe’s integration of image dehazing capabilities within the Camera Raw plugin of its renowned image editing application, Photoshop [10].

Recent years have witnessed rapid developments in self-driving vehicles and smart surveillance systems, where computer vision algorithms play crucial roles. Integrating image dehazing into these systems presents a key requirement, that is, processing speed. For instance, Bosch’s multi-purpose camera [11] is a system-on-a-chip (SoC) device designed for video-based driver assistance systems, and it can generate up to 45 frames per second (fps) at 2048 × 1280 resolution. If a dehazing algorithm with a processing speed of 10 fps were to be implemented, a bottleneck would arise, leading to significant performance loss. To maintain the high performance and smooth functioning of Bosch’s camera, the dehazing algorithm must handle images at a minimum speed of 25 or 30 fps, depending on whether the video encoding system is PAL or NTSC. This example highlights the critical importance of processing speed in real-time computer vision systems.

The following section summarizes the five-decade development of image dehazing, with a primary focus on daytime single-image approaches. It also highlights a trade-off between performance and algorithmic complexity. Sections 3 and 4 then detail the proposed solution based on multiprocessor system-on-a-chip (MPSoC) to balance this trade-off. Our contributions can be summarized as follows:

- We incorporate a self-calibrating feature, enabling the proposed algorithm to handle various haze conditions effectively.
- We present a real-time high-quality hardware implementation, facilitating the practical deployment of the proposed algorithm.

Experimental results are also presented to validate the real-time processing capability and performance, comparing the proposed solution with the base algorithm and state-of-the-art methods. Section 5 discusses future development directions and concludes the paper.

2. Image Dehazing Chronicle

Generally, image dehazing algorithms can be broadly categorized into two groups: heuristic and data-driven methods. These two categories differ in the origin of the utilized image features. Specifically, heuristic methods are grounded on handcrafted features discovered through engineering efforts. Conversely, data-driven methods focus on architecture design to learn the most representative features from the abundant data. This paper further classifies heuristic methods into enhancement-based and prior-based and data-driven methods into restoration-based and generation-based approaches. The following subsections chronicle major milestones in the development of single-image dehazing, along with high-impact studies exemplifying each individual category.

2.1. Heuristic Methods

2.1.1. Enhancement-Based Approach

The presence of haze causes atmospheric scattering and absorption, wherein part of the incoming light scatters directly into the camera's aperture, leading to increased brightness. The remaining light attenuates in the transmission medium before reaching the aperture, resulting in faintness. Consequently, hazy images exhibit poor visibility. To address this, enhancement-based methods aim to improve low-level features such as contrast and brightness. For example, Kim et al. [2] introduced the block-overlapped histogram equalization method, which spreads out highly populated intensities to enhance global contrast. This computationally inexpensive method is suitable for mobile phones and security cameras. However, for images with imperceptible background noise, this method may increase noise contrast, reducing the signal-to-noise ratio (SNR).

In line with this, Oakley and Satherley [3] proposed a physical-model-based contrast enhancement method to compensate for the adverse effects of a turbid atmosphere on digital images. They also noticed the SNR reduction problem and devised a temporal filter, but the problem persisted. Ancuti and Ancuti [12] adopted multiscale image fusion to alleviate the visibility reduction problem. They transformed a single input image using white balance and contrast enhancement to generate multiple variants of the input for image fusion. After that, they constructed Laplacian pyramids and conducted fusion using guidance weights derived from saliency, luminance, and chrominance. While multiscale image fusion guarantees dehazing results with fine details, it can hinder real-time hardware implementation due to up-sampling and down-sampling processes that require many frame buffers.

Similarly, Galdran [13] attempted to reverse the effects of atmospheric scattering and absorption through multiscale image fusion. A single input image undergoes artificial under-exposure and contrast enhancement to mitigate the haze-induced problems of brightness increase and contrast degradation. The resulting variants, expressed as Laplacian pyramids, are then input into the fusion process, where the weighting function is derived from the pixel-wise contrast and saturation maps, and while artificial under-exposure renders this method robust to noise amplification, it can also darken the dehazing results.

2.1.2. Prior-Based Approach

Enhancement-based methods focus on visibility restoration by manipulating low-level features, but they often overlook haze's impact on image degradation. To address this, researchers have modeled image formation in a turbid atmosphere using optical physics, with the most widely used model being the Koschmieder model.

$$\mathbf{I}(x) = \mathbf{J}(x)t(x) + \mathbf{A}[1 - t(x)], \quad (1)$$

where $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ denotes the hazy image, $\mathbf{J} \in \mathbb{R}^{H \times W \times 3}$ the clean image, $t \in \mathbb{R}^{H \times W}$ the transmission map, and $\mathbf{A} \in \mathbb{R}^{1 \times 1 \times 3}$ the global atmospheric light. H and W represent the height and width of images, and x denotes the spatial coordinates of pixels. The terms $\mathbf{J}(x)t(x)$ and $\mathbf{A}[1 - t(x)]$ correspond to the multiplicative and additive attenuation of the incoming light due to absorption and scattering. This model assumes a constant transmission map across color channels, whereas in reality, it is wavelength-dependent.

Another approach, prior-based methods, leverages prior knowledge to estimate the transmission map and global atmospheric light, then reverses Equation (1) to restore visibility. He et al. [4] proposed that in local image patches (excluding white and bright regions), pixels have extremely low intensities in at least one color channel, a concept known as the dark channel prior. For a local patch $\Omega(x)$ centered at x , this prior is represented by Equation (2). By substituting \mathbf{I} , \mathbf{A} , and t for \mathbf{J} , a direct relation between \mathbf{A} and t is established. He et al. [4] also suggested that the global atmospheric light corresponds to the brightest pixel in the top 0.1% of highest intensities in the dark channel.

$$\min_{y \in \Omega(x)} \left[\min_{c \in \{R, G, B\}} \mathbf{J}^c(y) \right] \approx 0. \quad (2)$$

Despite its simplicity, the method devised by He et al. [4] is highly effective, though it may cause color distortion in the sky region where the dark channel prior does not hold. Combining multiple priors can help address these limitations. For instance, Tang et al. [14] adopted random forest regression to estimate the transmission map from four image features: dark channel, contrast, saturation, and hue disparity, extracted at four different scales. They adapted the method of He et al. [4] by using the median (instead of the largest) of the top 0.1% of dark channel values for atmospheric light, improving robustness at the cost of a prolonged execution time.

The prior knowledge presented in the aforementioned dehazing solutions is verifiable with local image patches but not with global image context. Accordingly, Berman et al. [15] introduced a non-local prior, noting that colors in clean images form tight clusters in RGB space, spread throughout the image. They employed k -means clustering to identify these clusters and infer transmission values from their distance to the camera. The global atmospheric light estimation was similar to the method of He et al. [4]. This non-local prior is effective and versatile, as demonstrated in [16–18]. However, it shares common problems with other prior-based methods, such as a tendency to produce over-saturated dehazing results.

2.2. Data-Driven Methods

2.2.1. Restoration-Based Approach

To enhance the generalizability of dehazing algorithms, researchers have incorporated deep learning techniques, notably convolutional neural networks (CNNs). Cai et al. [6] proposed DehazeNet, which infers the transmission map from a single image. The approach employs a CNN to extract low-level features like contrast, saturation, and edge details. Maxout layers enhance feature robustness, and convolutional layers with different kernel sizes induce scale-invariant characteristics. Multiscale features undergo a max-pooling layer to enhance resilience against minor displacements in the input image. Finally, a bilateral ReLU performs nonlinear regression to estimate the transmission map. Cai et al. [6] utilized

the method of He et al. [4] to obtain the global atmospheric light, necessary for recovering clean images based on the Koschmieder model.

Haze-induced image degradation affects all red, green, and blue channels. However, Wang et al. [19] observed that it predominantly impacts the luminance channel. They developed a lightweight variant of DehazeNet to estimate the transmission map from the image's luminance, reducing computational costs while maintaining performance. Dudhane and Murala [20] extended the research of Cai et al. [6] and Wang et al. [19] by employing two DehazeNet-like networks to estimate two transmission maps in RGB and YCbCr color spaces, combining them with a fusion network to obtain the final transmission map. This method improved performance but increased computational costs. Recently, Sahu et al. [21] presented a dual-channel DehazeNet to improve the accuracy of transmission map estimation. To attain computational efficiency, they implemented their proposed model on an FPGA board, where the input images were downsampled to 32×32 pixels for real-time processing.

Ren et al. [22] proposed an alternative approach, which has also been widely referenced in subsequent studies. They devised a deep CNN that estimates the transmission map in a coarse-to-fine manner. They employed convolutional layers with large receptive fields to learn the coarse structure and layers with small receptive fields to refine the transmission map, ensuring smoothness while preserving discontinuities.

Despite their potential for learning complex and abstract patterns from images, the aforementioned methods solely utilized CNNs for estimating the transmission map. Additionally, the lack of real ground-truth data for training CNNs limits these methods, rendering them susceptible to the domain-shift problem.

2.2.2. Generation-Based Approach

The seminal work of Goodfellow et al. [23] on generative adversarial networks, coupled with the increasing adoption of the autoencoder architecture [24], has given rise to generation-based dehazing. Pan et al. [25] proposed a physics-based network involving hazy-clean generation followed by clean-hazy regeneration, using a separate discriminator to ensure consistency with the real input. The authors also incorporated the Koschmieder model to facilitate the regeneration. However, the network's capability is constrained by this physical model and may fail with complex phenomena.

In contrast to the method of Pan et al. [25], Liu et al. [26] improved upon previous work [27] with GridDehazeNet+, an enhanced multiscale network that dehazes images and is purely data-driven without relying on the Koschmieder model. GridDehazeNet+ processes images through pre-processing, multiscale image fusion, and post-processing stages. The multiscale processing employs a grid-like data flow with self-attention to combine data at different scales. Liu et al. [26] also addressed the domain-shift problem by utilizing intra-task knowledge transfer, training a teacher network with synthetic images and initializing a student network with its weights, then training with translated images via CycleGAN [28]. Anecdotally, using these translated images for model training is a provisional solution to the domain-shift problem, as they are still artificially generated.

Inspired by the concept of layer disentanglement [29], Li et al. [30] introduced an unsupervised network, which was trained to generate the transmission map and global atmospheric light in addition to the clean image. By reconstructing the hazy image using the Koschmieder model, the network can supervise itself during parameter optimization. This self-supervision capability significantly aids data preparation, though lacking clean image domain knowledge may hinder an optimal parameter search.

Recently, Xu et al. [8] introduced a U-Net-like network for video dehazing with a multiscale encoder and a prior-scene decoder. The multiscale encoder extracts feature maps at various scales, while the prior-scene decoder layers learn features related to the prior and scene. Recurrent features from adjacent video frames are aligned and aggregated to generate the clean image. The network, trained on synthesized hazy videos, remains susceptible to the domain-shift problem and fails to meet real-time processing requirements.

Wu et al. [9] sought to mitigate the domain-shift problem by incorporating diverse degradation types in hazy image synthesis, modifying the Koschmieder model to adjust light conditions, atmospheric light color bias, and JPEG compression effects. However, given that the distribution of the synthesized hazy images does not align with that of real hazy images, their proposed method remains a provisional solution. Sahu et al. [31] proposed Oval-Net, an encoder–decoder network with spatial and channel attention mechanisms, for end-to-end image dehazing. Oval-Net was trained using synthetic datasets, and the authors acknowledged that this could reduce the network’s reliability in real-world circumstances.

The primary concern with deep-learning-based methods is their high computational complexity. Even a simple network for image classification can contain millions of parameters, necessitating significant effort for hyper-parameter tuning and hindering fast and efficient implementation for widespread deployment. Chen et al. [32] addressed these challenges with a lightweight dehazing network using an autoencoder architecture, incorporating difference convolution to integrate low-level prior information and a content-guided attention mechanism for handling haze heterogeneity; while this network exhibited relative speed and efficiency, it still falls short of real-time high-quality image processing.

Most data-driven methods are trained on a mix of synthetic and real-world images. With the increasing prevalence of text-to-image models like StableDiffusion [33] and DALL-E 2 [34], this trend is expected to continue. Nonetheless, the inclusion of synthetic images may exacerbate the domain-shift problem, as discussed by Shumailov et al. [35]. Artificially generated images do not share a similar distribution with real-world images, potentially reducing network generalizability.

In summary, data-driven methods have demonstrated superior performance over heuristic approaches in various computer vision tasks. Nevertheless, their limitations, such as high computational cost and limited generalizability, may render them less favorable for practical applications.

2.3. Summary

Table 1 presents a summary of the daytime single-image dehazing methods discussed above. Generally, heuristic methods are computationally efficient but susceptible to noise, and their results tend to align with human perception. However, they may face challenges when applied to diverse circumstances. On the other hand, data-driven methods offer improved generalizability but come with a higher computational cost, and their results often align better with quantitative assessment metrics. Notably, nearly all data-driven methods are susceptible to the domain-shift problem due to the lack of real-world training images.

Table 1. Summary of daytime single-image dehazing chronicle.

Category	Representative Studies	Pros and Cons
Heuristic	Enhancement-based [2,3,12,13]	Pros Low computational cost Subjectively favoring results
	Prior-based [4,14,15]	Cons Noise amplification Lack of generalizability
Data-driven	Restoration-based [6,19,20,22]	Pros Improved generalizability Quantitatively favoring results
	Generation-based [8,25,26,30,32]	Cons High computational cost Domain-shift problem

More importantly, data-driven methods often require graphics processing units (GPUs) for model inference. Since this computing platform is power-consuming and expensive,

it is unsuitable for implementation on edge devices, such as CCTVs or cameras mounted on autonomous driving vehicles. In sharp contrast, the proposed MPSoC-based solution presented in the following section is fast and compact, occupying less than one-fifth of the hardware resources available on a mid-size FPGA device (XCZU7EV-2FFVC1156), as demonstrated in Section 4.2. This makes the proposed MPSoC-based solution a preferable option over data-driven methods for real-time high-quality image dehazing.

3. Proposed Algorithm

As data-driven methods are not yet ready for widespread deployment, this paper presents an alternative for real-time high-quality single-image dehazing: a symmetric MPSoC-based solution that balances the trade-off between dehazing performance and computational complexity. Building upon our previous work of linear-time single-image dehazing [36], the proposed algorithm incorporates the following features (as illustrated in Figure 1):

- A self-calibrating feature that enables the algorithm to handle different haze conditions effectively.
- A real-time high-quality hardware implementation that facilitates the practical deployment of the proposed algorithm.

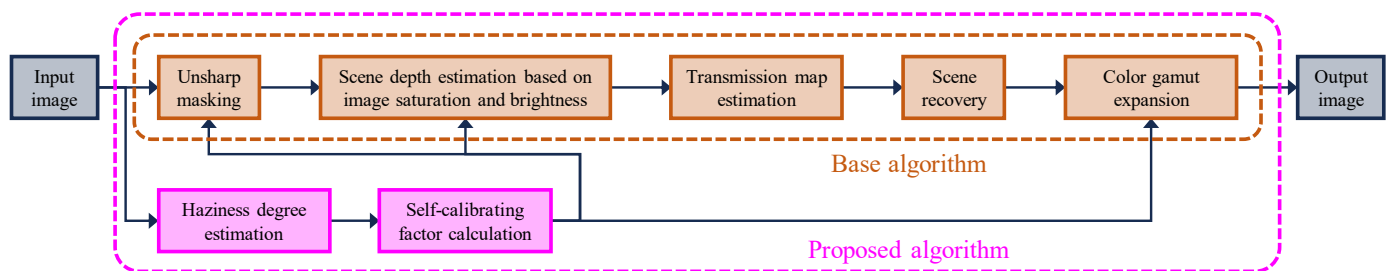


Figure 1. Block diagram of the proposed algorithm.

3.1. Base Algorithm

In [36], we presented an $\mathcal{O}(N)$ dehazing method, where N denotes the number of image pixels. This method enhances the visibility of hazy images through several steps. Initially, a pre-processing step involving unsharp masking is applied to the input image to enhance edge details based on local image statistics. Next, image visibility is restored using a dehazing step grounded on the improved color attenuation prior. However, this dehazing step may introduce artifacts like dynamic range reduction. To address this, a post-processing step, namely color gamut expansion, is employed to ensure an artifact-free output. Interested readers are referred to [36] for a more comprehensive description.

Let \mathcal{P} , \mathcal{D} , and \mathcal{H} denote the pre-processing, dehazing, and post-processing stages of the base algorithm. The clean image \mathbf{J} is derived from the hazy image \mathbf{I} as follows:

$$\mathbf{J} = \mathcal{H}\{\mathcal{D}[\mathcal{P}(\mathbf{I})]\}, \quad (3)$$

where spatial coordinates are omitted for clarity. The responses of these three stages to input images are fixed, irrespective of whether images are affected by haze. Hence, the following subsection outlines our contribution in adopting a haziness degree estimator [37] to make \mathcal{P} , \mathcal{D} , and \mathcal{H} aware of haze conditions. More precisely, we introduce a self-calibrating weight ω into \mathcal{P} , \mathcal{D} , and \mathcal{H} to restore the clean image as $\mathbf{J} = \mathcal{H}\{\omega, \mathcal{D}[\omega, \mathcal{P}(\omega, \mathbf{I})]\}$. Depending on the haze condition of the input image, the value of ω varies, thus enabling the fine-tuning of all three processing stages for appropriate enhancement.

In [36], we demonstrated that the base algorithm achieves performance comparable to data-driven methods, such as those proposed by Cai et al. [6] and Ren et al. [22], while exhibiting significantly lower computational costs. However, real-time processing requirements through software implementation remain challenging. The fastest implementation,

representing the base algorithm, processes only ten 640×480 frames per second (fps), falling short of the desired 25 fps.

The subsequent subsections focus on two main aspects. Firstly, efforts are made to incorporate a self-calibrating feature into the base algorithm to enhance performance further. Secondly, a comparative evaluation is conducted to assess the effectiveness of the proposed improvements. Additionally, Section 4 presents an MPSoC-based solution to address the real-time processing constraint.

3.2. Self-Calibration on Haze Conditions

The Koschmieder model describes the transmission map $t(x)$ as an exponential function of the scene depth $d(x)$, denoted as $t(x) = \exp[-\beta \cdot d(x)]$, where β represents the atmospheric scattering coefficient. This implies that the haze distribution depends on scene depth, allowing the dehazing algorithm to handle various types of haze, from mild to dense. In a prior study [38], we introduced a framework for generating a piece-wise linear weight using the haziness degree estimator [37]. This weight is combined with the scene depth in a multiplicative manner to address different scenarios:

- **Haze-free images.** The weight is set to zero, zeroing the scene depth. Consequently, $t(x) = 1$ is achieved throughout the image, meaning that no image dehazing is performed.
- **Mildly-to-moderately hazy images.** The weight assumes a value ω_e , where $0 < \omega_e < 1$, based on the haziness degree estimate, reducing the dehazing power to prevent artifacts.
- **Densely hazy images.** The weight is set to one, imposing no constraints on the scene depth, allowing maximum dehazing power.

By incorporating this adaptive weight, the base algorithm can effectively adapt to various haze conditions, improving results for different types of hazy images. Figure 2a illustrates this weighting scheme, where ω and ρ represent the weight and haziness degree estimate, respectively. The haziness degree range is divided into three regions using two predefined parameters, ρ_1 and ρ_2 . The weighting scheme is expressed as follows:

$$\omega = \begin{cases} 0 & \rho < \rho_1 \\ \frac{\rho - \rho_1}{\rho_2 - \rho_1} & \rho_1 \leq \rho \leq \rho_2 \\ 1 & \rho > \rho_2 \end{cases} \quad (4)$$

In [38], evaluation results indicated that dehazing performance for densely hazy images was unimpressive, suggesting that greater dehazing power might improve results. Consequently, in this study, we have modified the original weighting scheme by allowing the weight (ω) to extend beyond the range $[0, 1]$, up to a predefined value of \mathcal{W} , as illustrated in Figure 2b. This modification enables the algorithm to effectively “see” through a thicker haze, surpassing the capabilities of prior-based dehazing methods. The proposed weighting scheme is expressed as follows:

$$\omega = \begin{cases} 0 & \rho < \rho_1 \\ \frac{\rho - \rho_1}{\rho_2 - \rho_1} & \rho_1 \leq \rho \leq \rho_2 \\ \left(\frac{\mathcal{W} - 1}{1 - \rho_2}\right)\rho + \frac{1 - \mathcal{W}\rho_2}{1 - \rho_2} & \rho > \rho_2 \end{cases} \quad (5)$$

Figure 2 illustrates a comparison between the original weight presented in [38] and our proposed weight to highlight their differences. It features a densely hazy image from the IVC dataset [39] and showcases two dehazing results obtained using the two weights, respectively. Parameters ρ_1 and ρ_2 are set to 0.8811 and 0.9344, as described in [40], and \mathcal{W} is fixed at 1.2. Subjective evaluation shows that the dehazing result in Figure 2d is less favorable compared to the result with our proposed weight in Figure 2e.

Our contribution extends beyond the weighting scheme, including how the self-calibrating weight is incorporated into the algorithm. In [38], the self-calibrating weight was applied to both the dehazing and post-processing stages, while the pre-processing stage remained unchanged. The pre-processing stage focused on white-balancing the input image to skip the estimation of the global atmospheric light \mathbf{A} (in [38], the global atmospheric light was set to a fixed value of $\{1, 1, 1\}$, under the assumption that image intensities were normalized within the range $[0, 1]$). In contrast, the proposed algorithm uses unsharp masking in the pre-processing stage to enhance distant edge details obscured by haze. Consequently, we have also equipped this pre-processing stage with the self-calibrating weight to prevent overshooting in haze-free images.

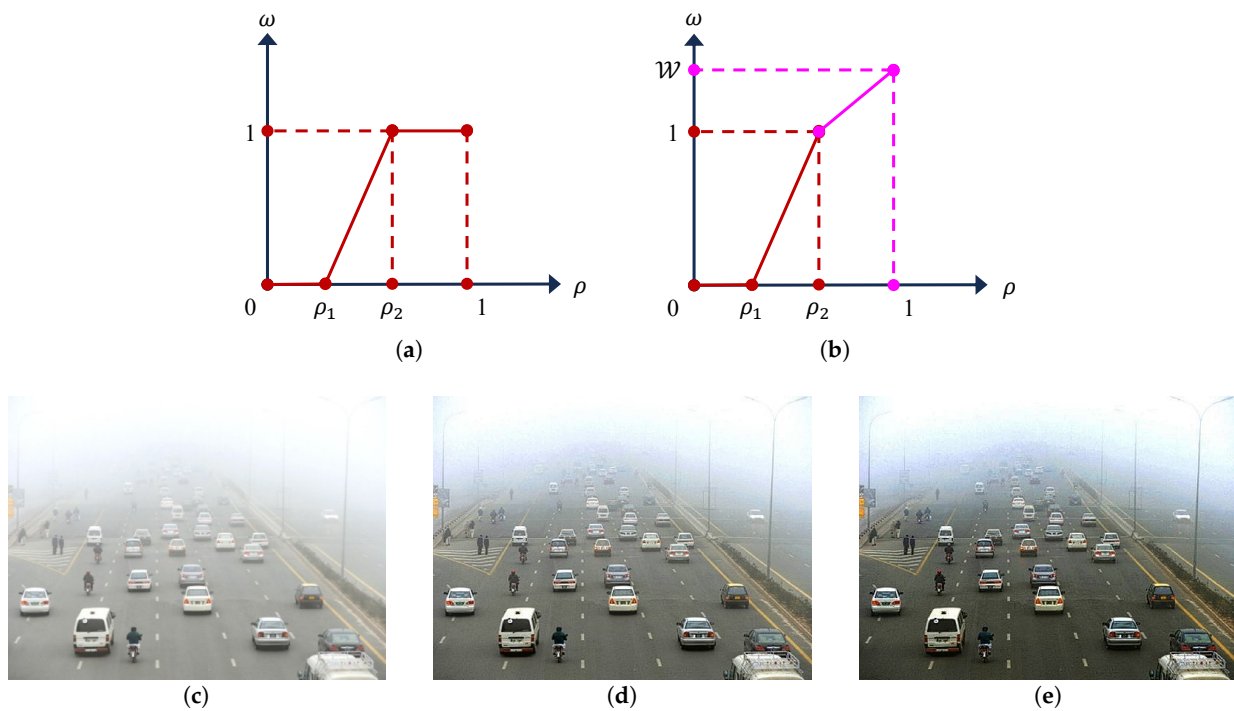


Figure 2. Illustration of piece-wise linear weights for incorporating the self-calibrating feature. (a) The original weight presented in [38]. (b) Our proposed weight. (c) A densely hazy image. (d) Dehazing result with the original weight. (e) Dehazing result with the proposed weight.

3.3. Objective Evaluation

To validate the performance of the proposed algorithm, we conducted a comparative analysis against four methods, including the base algorithm and those proposed by Cai et al. [6], Liu et al. [27], and Li et al. [30]. We used five public datasets for evaluation: FRIDA2 [41], D-HAZY [42], O-HAZE [43], I-HAZE [44], and Dense-Haze [45]. The benchmark methods have been introduced in Section 2.

The FRIDA2 dataset consists of 320 computer-rendered images of road scenes, with 66 haze-free and 264 hazy images representing four distinct haze conditions. D-HAZY is another synthetic dataset containing 1472 pairs of indoor hazy/clean images, where haze was synthesized using scene depth information from a Microsoft Kinect camera. In contrast, O-HAZE, I-HAZE, and Dense-Haze are real-world datasets consisting of 45, 30, and 55 image pairs, respectively, depicting outdoor, indoor, and both outdoor and indoor scenes.

To assess the experimental results, we employed two metrics: feature similarity extended to color images (FSIMc) [46] and tone-mapped image quality index (TMQI) [47]. Both metrics provide scores ranging from zero to one, where higher scores indicate better results. The obtained FSIMc and TMQI scores for each dataset, along with their average scores, are presented in Table 2.

The results demonstrate that the proposed algorithm, enhanced with the new self-calibrating weighting scheme, consistently outperforms the base algorithm in all test scenarios. Additionally, it is also ranked higher than the other three data-driven benchmark methods. This result suggests that even though the proposed method is a heuristic approach, it effectively addresses the limitation of limited generalizability, thanks to the new weighting scheme. With its efficacy confirmed, the next section will introduce a corresponding hardware accelerator to enhance its practical usability.

Table 2. Objective evaluation using feature similarity extended to color images (FSIMc) and tone-mapped image quality index (TMQI). The best results are highlighted in bold.

		Dataset	FRIDA2 [41]	D-HAZY [42]	O-HAZE [43]	I-HAZE [44]	Dense-Haze [45]	Overall
Method								
FSIMc	Cai et al. [6]		0.7963	0.8874	0.7865	0.8482	0.5573	0.7725
	Liu et al. [27]		0.8003	0.8747	0.8030	0.7416	0.5564	0.7552
	Li et al. [30]		0.7849	0.7383	0.6997	0.7564	0.5763	0.7111
	Base algorithm [36]		0.8016	0.8763	0.8112	0.8586	0.5728	0.7807
	Proposed algorithm		0.8029	0.8733	0.8265	0.8677	0.5799	0.7846
TMQI	Cai et al. [6]		0.7366	0.7966	0.8413	0.7598	0.5723	0.7312
	Liu et al. [27]		0.6970	0.7938	0.8267	0.6107	0.5196	0.6896
	Li et al. [30]		0.7176	0.6817	0.6566	0.6936	0.5107	0.6520
	Base algorithm [36]		0.7242	0.7841	0.8951	0.8204	0.5921	0.7354
	Proposed algorithm		0.7244	0.7790	0.8913	0.8101	0.6040	0.7357

4. MPSoC-Based Solution

4.1. Real-Time Hardware Implementation

Previous studies [32,36] have reported that software implementations of dehazing algorithms are unable to achieve a processing speed of at least 25 fps, failing to meet real-time processing requirements. This observation underscores the critical need for hardware implementation. In this study, we develop a hardware accelerator for the proposed algorithm using Verilog HDL [48] (IEEE Standard 1364-2005) and validate its performance on a Zynq UltraScale+ MPSoC ZCU106 Evaluation Kit [49].

Before implementing the hardware accelerator, let us revisit the block diagram in Figure 1. It is important to note that all operations in the base algorithm, including unsharp masking, image dehazing, and color gamut expansion, are pixel-wise. In contrast, our additions to the base algorithm, involving haziness degree estimation and self-calibrating factor calculation, are performed on a per-frame basis. This difference poses a challenging problem in synchronizing data flows within the proposed algorithm. Delaying unsharp masking, image dehazing, and color gamut expansion until the haziness degree estimate becomes available is impractical, as it leads to flickering issues.

To address this problem, we leveraged the high similarity between consecutive video frames, a common characteristic in various video types resulting from the natural continuity of motion and scenes in real-world scenarios. Numerous video processing and analysis techniques, such as motion estimation, video stabilization, object tracking, and video compression, have effectively exploited this characteristic.

Figure 3 shows a plot of structural similarity values [50] for the initial 300 frames of a video. The plot demonstrates that each frame exhibits a strong resemblance to its preceding and following frames, as indicated by the red-dotted oval, except during abrupt video changes highlighted by the blue-dotted and pink-dotted ovals. Given the infrequency of these scene changes, it is feasible to compute the haziness degree estimate for a specific frame and apply the computed value to the subsequent frame. This approach not only addresses the synchronization problem but also significantly reduces the required hardware resources for implementation.

After addressing the synchronization problem, the proposed algorithm can be readily implemented at the register-transfer level using standard design techniques. It is partitioned into blocks similar to the block diagram in Figure 1. By exploiting the pipeline parallelism, each block corresponds to a processing stage in the pipeline, allowing simultaneous processing of pixels from the previous stage, thus increasing the throughput. The resulting hardware accelerator is then encapsulated by an interface circuit to adhere to the AXI bus communication protocol [51]. Our interface circuit supports a double-buffering scheme, enabling the accelerator to seamlessly process the input video stream. Xilinx Vivado v2019.1 [52] was employed to develop a hardware intellectual property (IP) and program the Zynq UltraScale+ MPSoC ZCU106 Evaluation Kit.

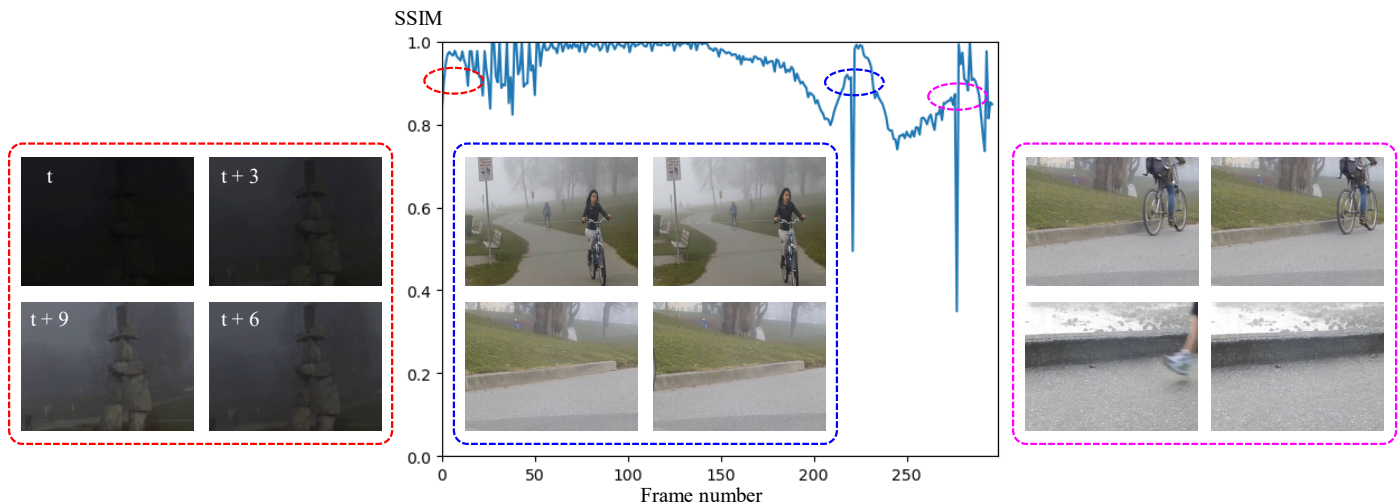


Figure 3. Plot of structural similarity (SSIM) values of the first 300 frames in a video. Red-dotted oval indicates frames of similar SSIM values, while blue-dotted and pink-dotted ovals indicate frames of abrupt changes in SSIM values.

Figure 4 provides an overview of our MPSoC-based solution. The input video stream is processed by a verification platform running on a host computer. This platform acts as an intermediary between our MPSoC-based solution and users. It receives, packages, and sends commands received from users, as well as video data, to our hardware IP within the evaluation kit. The kit features a Zynq UltraScale+ MPSoC device, comprising a quad-core ARM processor, a dual-core real-time processor, a graphics-processing unit, and a mid-size FPGA device (XCZU7EV-2FFVC1156) [49]. The ARM processor is referred to as the processing system (PS), and the FPGA device is referred to as the programmable logic (PL). We have developed an application called the hardware controller, which runs on the PS and is responsible for interfacing our hardware IP (located in the PL) with the outside world.

In this setup, the verification platform acquires the video stream from a camera and gathers user-inputted data through its graphical user interface. Following a hand-shaking process, the platform forwards the collected data to the hardware controller in the PS, which in turn relays the received data to the hardware IP in the PL. Subsequently, the hardware IP processes the data and generates an interrupt signal upon completion. The hardware controller acknowledges the signal and transmits the processed data back to the verification platform, where the input–output data are displayed side-by-side to users for ease of verification.

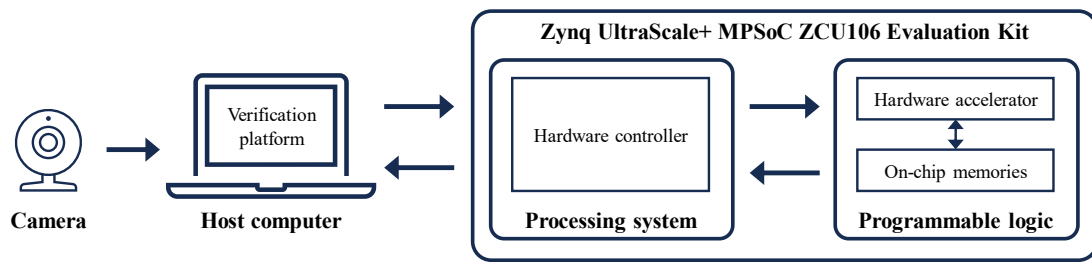


Figure 4. Overview of the proposed MPSoC-based solution.

4.2. Hardware Implementation Results

We utilized Xilinx Vivado v2019.1 [52] to synthesize the proposed hardware IP on the FPGA device. The implementation results, as summarized in Table 3, demonstrate that our hardware IP occupies only a modest portion of the available hardware resources. Specifically, it consumes 9.95% of slice registers, 19.86% of slice look-up tables (LUTs), and 17.47% of block RAMs (BRAMs). The FPGA device used in this study belongs to the Zynq UltraScale+ family, which features UltraRAMs and BRAMs. However, the proposed hardware accelerator does not contain any frame buffers, only line memories for filtering operations. Thus, BRAMs are adequate, leaving UltraRAMs available for other applications requiring frame buffers.

Table 3. Hardware implementation results for the proposed algorithm. LUT stands for look-up table, and the symbol # denotes quantities.

Xilinx Vivado v2019.1			
Device	XCZU7EV-2FFVC1156		
Slice Logic Utilization	Available	Used	Utilization
Slice registers (#)	460,800	45,832	9.95%
Slice LUTs (#)	230,400	45,761	19.86%
BRAMs	312	54.5	17.47%
Minimum period		2.81 ns	
Maximum frequency		356.51 MHz	

Furthermore, Table 3 shows that the proposed hardware accelerator can operate with a minimum clock period of 2.81 nanoseconds, allowing it to handle up to 356.51 megapixels per second. The maximum processing speed (S_{\max}) in fps for a given frame's resolution of $H \times W$ can be calculated as follows:

$$S_{\max} = \frac{f_{\max}}{(H + VB)(W + HB)}, \quad (6)$$

where f_{\max} represents the maximum frequency reported in Table 3, and VB and HB denote the vertical and horizontal blank periods, respectively. Table 4 presents the S_{\max} values for various resolutions, ranging from Full HD (1920×1080) to DCI 4K (4096×2160), demonstrating that the proposed hardware IP exceeds the real-time processing requirement. For DCI 4K resolution, it achieves a maximum processing speed of 40.27 fps, making it highly suitable for real-world computer vision systems, irrespective of the color encoding scheme employed.

Moreover, we conducted a comparative assessment of the proposed accelerator with existing designs for single-image dehazing [53–55]. Park and Kim [53] and Zhang and Zhao [54] presented their own approaches to implementing the method of He et al. [4]. Specifically, they explored alternative methods to estimate the global atmospheric light more cost-effectively. For instance, Park and Kim [53] divided the image into 12 non-overlapping regions and searched for atmospheric light candidates in each region. Subsequently, they selected the brightest pixel among the candidates as the

atmospheric light. Meanwhile, Zhang and Zhao [54] approximated the atmospheric light as the largest pixel in the locally filtered image (using a minimum filter).

Table 4. Maximum processing speeds in frames per second for various image resolutions. The symbol # denotes quantities.

Standard	Resolution	Required Clock Cycles (#)	Processing Speed (S_{\max})
Full HD	1920 × 1080	2,076,601	171.68
Quad HD	2560 × 1440	3,690,401	96.60
UW4K	3840 × 1600	6,149,441	57.97
4K UHD TV	3840 × 2160	8,300,401	42.95
DCI 4K	4096 × 2160	8,853,617	40.27

Table 5 illustrates that our hardware IP has the smallest footprint in terms of slice registers and LUTs while achieving the fastest processing speed. Notably, to the best of our knowledge, the proposed hardware IP, along with our previous design in [55], are the sole two hardware implementations equipped with the self-calibrating feature. Regarding digital signal processors (DSPs), they tend to be costly and are specifically designed for computationally intensive tasks, such as matrix multiplication in CNNs. Given that image dehazing frequently serves as a pre-processing step in high-level computer vision systems, it is preferable to reserve DSPs for more complex tasks like object recognition and localization. In this context, both of our previous and proposed designs excel by eschewing the use of DSPs. Through minimal resource utilization and objective evaluation, the proposed MPSoC-based solution achieves a balance between dehazing performance and computational complexity, hence termed “symmetric”.

Nonetheless, it is essential to acknowledge that our design necessitates a considerable amount of memory, primarily utilized as line memories in filtering operations. Considering the significant impact of these operations on the base algorithm’s performance, eliminating them is not a viable option. In future studies, we will explore solutions to reduce memory requirements without compromising performance.

Table 5. Comparison with contemporary hardware accelerators for single-image dehazing. NA stands for not available, and the symbol # denotes quantities.

Hardware Utilization	Park and Kim [53]	Zhang and Zhao [54]	Lee et al. [55]	Proposed Design
Registers (#)	53,400	NA	53,216	45,832
LUTs (#)	64,000	NA	49,799	45,761
DSPs (#)	42	NA	0	0
Memory (Mbits)	3.2	NA	1.4	2.0
Maximum frequency (MHz)	88.70	116.00	271.37	356.51
Maximum video resolution	SVGA	Quad HD	DCI 4K	DCI 4K
Self-calibrating feature	Unequipped	Unequipped	Equipped	Equipped

5. Conclusions

In this paper, we introduce a symmetric MPSoC-based solution to address the growing demand for real-time high-quality image dehazing. Our proposed method balances the trade-off between dehazing performance and computational complexity. It enhances the base algorithm by incorporating a self-calibrating feature, enabling efficient handling of various haze conditions. Furthermore, we have improved the piece-wise linear weighting scheme to enhance haze removal under dense-haze conditions. Subsequently, we have designed a corresponding hardware accelerator using Verilog HDL and verified its effectiveness against existing implementations.

However, we have identified three main limitations in our proposed solution. Firstly, it is inefficient in memory usage due to the extensive utilization of filtering operations. As these operations are crucial to the proposed algorithm’s performance, further refinement of the design demands substantial effort. Secondly, the proposed algorithm relies on several parameters that necessitate careful fine-tuning for optimal performance, which is a

laborious and time-consuming process. Finally, there is no encryption applied to the image data, posing a security risk. We defer the resolution of these three challenging problems to future research endeavors.

Author Contributions: Conceptualization, B.K.; methodology, B.K. and D.N.; software, D.N.; validation, D.N.; formal analysis, D.N.; investigation, D.N.; data curation, D.N.; writing—original draft preparation, D.N.; writing—review and editing, B.K. and D.N.; visualization, D.N.; supervision, B.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2023R1A2C1004592).

Data Availability Statement: Data are available in a publicly accessible repository. The data presented in this study are openly available in [41–45].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Vincent, R. An ERTS Multispectral Scanner experiment for mapping iron compounds. In Proceedings of the Eight International Symposium on Remote Sensing of Environment, Ann Arbor, MI, USA, 2–6 October 1972; pp. 1239–1247.
2. Kim, T.K.; Paik, J.K.; Kang, B.S. Contrast enhancement system using spatially adaptive histogram equalization with temporal filtering. *IEEE Trans. Consum. Electron.* **1998**, *44*, 82–87. [[CrossRef](#)]
3. Oakley, J.P.; Satherley, B.L. Improving image quality in poor visibility conditions using a physical model for contrast degradation. *IEEE Trans. Image Process.* **1998**, *7*, 167–179. [[CrossRef](#)] [[PubMed](#)]
4. He, K.; Sun, J.; Tang, X. Single Image Haze Removal Using Dark Channel Prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 2341–2353. [[CrossRef](#)] [[PubMed](#)]
5. Zhu, Q.; Mai, J.; Shao, L. A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior. *IEEE Trans. Image Process.* **2015**, *24*, 3522–3533. [[CrossRef](#)]
6. Cai, B.; Xu, X.; Jia, K.; Qing, C.; Tao, D. DehazeNet: An End-to-End System for Single Image Haze Removal. *IEEE Trans. Image Process.* **2016**, *25*, 5187–5198. [[CrossRef](#)]
7. Li, R.; Pan, J.; He, M.; Li, Z.; Tang, J. Task-Oriented Network for Image Dehazing. *IEEE Trans. Image Process.* **2020**, *29*, 6523–6534. [[CrossRef](#)]
8. Xu, J.; Hu, X.; Zhu, L.; Dou, Q.; Dai, J.; Qiao, Y.; Heng, P.A. Video Dehazing via a Multi-Range Temporal Alignment Network with Physical Prior. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 18053–18062. [[CrossRef](#)]
9. Wu, R.; Duan, Z.; Guo, C.; Chai, Z.; Li, C. RIDCP: Revitalizing Real Image Dehazing via High-Quality Codebook Priors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 22282–22291. [[CrossRef](#)]
10. Adobe. Vignette, Grain, and Dehaze Effects in Camera Raw. Available online: <https://helpx.adobe.com/camera-raw/using/vignette-grain-effects-camera-raw.html> (accessed on 6 September 2023).
11. Bosch. Multi Purpose Camera: Combination of Classic Cutting Edge Computer Vision Algorithms and Artificial Intelligence Methods. Available online: <https://www.bosch-mobility.com/en/solutions/camera/multi-purpose-camera/> (accessed on 12 June 2023).
12. Ancuti, C.O.; Ancuti, C. Single image dehazing by multi-scale fusion. *IEEE Trans. Image Process.* **2013**, *22*, 3271–3282. [[CrossRef](#)] [[PubMed](#)]
13. Galdran, A. Image dehazing by artificial multiple-exposure image fusion. *Signal Process.* **2018**, *149*, 135–147. [[CrossRef](#)]
14. Tang, K.; Yang, J.; Wang, J. Investigating Haze-Relevant Features in a Learning Framework for Image Dehazing. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2995–3002. [[CrossRef](#)]
15. Berman, D.; Treibitz, T.; Avidan, S. Single Image Dehazing Using Haze-Lines. *IEEE Trans. Pattern. Anal. Mach. Intell.* **2020**, *42*, 720–734. [[CrossRef](#)]
16. Hu, H.M.; Guo, Q.; Zheng, J.; Wang, H.; Li, B. Single Image Defogging Based on Illumination Decomposition for Visual Maritime Surveillance. *IEEE Trans. Image Process.* **2019**, *28*, 2882–2897. [[CrossRef](#)]
17. Afridi, I.U.; Bashir, T.; Khattak, H.A.; Khan, T.M.; Imran, M. Degraded image enhancement by image dehazing and Directional Filter Banks using Depth Image based Rendering for future free-view 3D-TV. *PLoS ONE* **2019**, *14*, e0217246. [[CrossRef](#)]
18. Berman, D.; Levy, D.; Avidan, S.; Treibitz, T. Underwater Single Image Color Restoration Using Haze-Lines and a New Quantitative Dataset. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**. [[CrossRef](#)] [[PubMed](#)]
19. Wang, A.; Wang, W.; Liu, J.; Gu, N. AIPNet: Image-to-Image Single Image Dehazing with Atmospheric Illumination Prior. *IEEE Trans. Image Process.* **2019**, *28*, 381–393. [[CrossRef](#)] [[PubMed](#)]

20. Dudhane, A.; Murala, S. RYF-Net: Deep Fusion Network for Single Image Haze Removal. *IEEE Trans. Image Process.* **2020**, *29*, 628–640. [[CrossRef](#)] [[PubMed](#)]
21. Sahu, G.; Seal, A.; Yazidi, A.; Krejcar, O. A Dual-Channel Dehaze-Net for Single Image Dehazing in Visual Internet of Things Using PYNQ-Z2 Board. *IEEE Trans. Autom. Sci. Eng.* **2024**, *21*, 305–319. [[CrossRef](#)]
22. Ren, W.; Pan, J.; Zhang, H.; Cao, X.; Yang, M.H. Single Image Dehazing via Multi-scale Convolutional Neural Networks with Holistic Edges. *Int. J. Comput. Vis.* **2020**, *128*, 240–259. [[CrossRef](#)]
23. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the 27th International Conference on Neural Information Processing Systems—Volume 2, Montreal, ON, Canada, 8–13 December 2014; pp. 2672–2680.
24. Kramer, M.A. Nonlinear principal component analysis using autoassociative neural networks. *AIChE J.* **1991**, *37*, 233–243. [[CrossRef](#)]
25. Pan, J.; Dong, J.; Liu, Y.; Zhang, J.; Ren, J.; Tang, J.; Tai, Y.W.; Yang, M.H. Physics-Based Generative Adversarial Models for Image Restoration and Beyond. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 2449–2462. [[CrossRef](#)]
26. Liu, X.; Shi, Z.; Wu, Z.; Chen, J.; Zhai, G. GridDehazeNet+: An Enhanced Multi-Scale Network With Intra-Task Knowledge Transfer for Single Image Dehazing. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 870–884. [[CrossRef](#)]
27. Liu, X.; Ma, Y.; Shi, Z.; Chen, J. GridDehazeNet: Attention-Based Multi-Scale Network for Image Dehazing. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7313–7322. [[CrossRef](#)]
28. Zhu, J.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2242–2251. [[CrossRef](#)]
29. Li, B.; Gou, Y.; Liu, J.Z.; Zhu, H.; Zhou, J.T.; Peng, X. Zero-Shot Image Dehazing. *IEEE Trans. Image Process.* **2020**, *29*, 8457–8466. [[CrossRef](#)]
30. Li, B.; Gou, Y.; Gu, S.; Liu, J.Z.; Zhou, J.T.; Peng, X. You Only Look Yourself: Unsupervised and Untrained Single Image Dehazing Neural Network. *Int. J. Comput. Vis.* **2021**, *129*, 1754–1767. [[CrossRef](#)]
31. Sahu, G.; Seal, A.; Jaworek-Korjakowska, J.; Krejcar, O. Single Image Dehazing via Fusion of Multilevel Attention Network for Vision-Based Measurement Applications. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 4503415. [[CrossRef](#)]
32. Chen, Z.; He, Z.; Lu, Z.M. DEA-Net: Single image dehazing based on detail-enhanced convolution and content-guided attention. *arXiv* **2023**, arXiv:2301.04805. [[CrossRef](#)]
33. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. *arXiv* **2022**, arXiv:2112.10752. [[CrossRef](#)]
34. Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; Chen, M. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv* **2022**, arXiv:2204.06125. [[CrossRef](#)]
35. Shumailov, I.; Shumaylov, Z.; Zhao, Y.; Gal, Y.; Papernot, N.; Anderson, R. The Curse of Recursion: Training on Generated Data Makes Models Forget. *arXiv* **2023**, arXiv:2305.17493. [[CrossRef](#)]
36. Ngo, D.; Lee, G.D.; Kang, B. Single Image Dehazing With Unsharp Masking and Color Gamut Expansion. *IEEE Access* **2022**, *10*, 102462–102474. [[CrossRef](#)]
37. Ngo, D.; Lee, G.D.; Kang, B. Haziness Degree Evaluator: A Knowledge-Driven Approach for Haze Density Estimation. *Sensors* **2021**, *21*, 3896. [[CrossRef](#)] [[PubMed](#)]
38. Ngo, D.; Lee, S.; Kang, U.J.; Ngo, T.M.; Lee, G.D.; Kang, B. Adapting a Dehazing System to Haze Conditions by Piece-Wisely Linearizing a Depth Estimator. *Sensors* **2022**, *22*, 1957. [[CrossRef](#)]
39. Ma, K.; Liu, W.; Wang, Z. Perceptual evaluation of single image dehazing algorithms. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 3600–3604. [[CrossRef](#)]
40. Ngo, D.; Lee, S.; Lee, G.D.; Kang, B. Automating a Dehazing System by Self-Calibrating on Haze Conditions. *Sensors* **2021**, *21*, 6373. [[CrossRef](#)]
41. Tarel, J.; Hautiere, N.; Caraffa, L.; Cord, A.; Halmaoui, H.; Gruyer, D. Vision Enhancement in Homogeneous and Heterogeneous Fog. *IEEE Intell. Transp. Syst. Mag.* **2012**, *4*, 6–20. [[CrossRef](#)]
42. Ancuti, C.; Ancuti, C.; Vleeschouwer, C. D-HAZY: A dataset to evaluate quantitatively dehazing algorithms. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 2226–2230. [[CrossRef](#)]
43. Ancuti, C.; Ancuti, C.; Timofte, R.; Vleeschouwer, C. O-HAZE: A Dehazing Benchmark with Real Hazy and Haze-Free Outdoor Images. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 867–8678. [[CrossRef](#)]
44. Ancuti, C.; Ancuti, C.; Timofte, R.; De Vleeschouwer, C. I-HAZE: A dehazing benchmark with real hazy and haze-free indoor images. *arXiv* **2018**, arXiv:1804.05091.
45. Ancuti, C.; Ancuti, C.; Sbert, M.; Timofte, R. Dense-Haze: A Benchmark for Image Dehazing with Dense-Haze and Haze-Free Images. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 1014–1018. [[CrossRef](#)]

46. Zhang, L.; Zhang, L.; Mou, X.; Zhang, D. FSIM: A Feature Similarity Index for Image Quality Assessment. *IEEE Trans. Image Process.* **2011**, *20*, 2378–2386. [[CrossRef](#)]
47. Yeganeh, H.; Wang, Z. Objective Quality Assessment of Tone-Mapped Images. *IEEE Trans. Image Process.* **2013**, *22*, 657–667. [[CrossRef](#)] [[PubMed](#)]
48. *Std 1364-2005*; IEEE Standard for Verilog Hardware Description Language. Revision of IEEE Std 1374-2001; IEEE: New York, NY, USA, 2006; pp. 1–590. [[CrossRef](#)]
49. Xilinx. ZCU106 Evaluation Board: User Guide (UG1244). Available online: <https://docs.xilinx.com/v/u/en-US/ug1244-zcu106-eval-bd> (accessed on 25 July 2023).
50. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
51. ARM. AMBA. Available online: <https://developer.arm.com/Architectures/AMBA> (accessed on 8 September 2023).
52. Xilinx. Vivado Design Suite User Guide: Release Notes, Installation, and Licensing (UG973). Available online: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_1/ug973-vivado-release-notes-install-license.pdf (accessed on 31 December 2021).
53. Park, Y.; Kim, T.H. A video dehazing system based on fast airlight estimation. In Proceedings of the 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Montreal, ON, Canada, 14–16 November 2017; pp. 779–783. [[CrossRef](#)]
54. Zhang, B.; Zhao, J. Hardware Implementation for Real-Time Haze Removal. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2017**, *25*, 1188–1192. [[CrossRef](#)]
55. Lee, S.; Ngo, D.; Kang, B. Design of an FPGA-Based High-Quality Real-Time Autonomous Dehazing System. *Remote Sens.* **2022**, *14*, 1852. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.