EasyChair Preprint

№ 10762

Lightweight Separable Convolutional Dehazing Network to Mobile FPGA

Xinrui Ju, Wei Wang and Xin Xu

August 22, 2023

Lightweight Separable Convolutional Dehazing Network to Mobile FPGA

Xinrui Ju[1], Wei Wang*[1,2][0000−0002−3733−3939], and Xin Xu[1,2]

[1] School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, 430081, China
[2] Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, Wuhan, 430081, China
wangwei8@wust.edu.cn

Abstract. The advancement of deep learning has significantly increased the efficiency of picture dehazing techniques. Convolutional neural networks can't, however, be implemented on portable FPGA devices because to their high computing, storage, and energy needs. In this paper, we propose a generic solution for image dehazing from CNN models to mobile FPGAs. The proposed solution designs lightweight network using depth-wise separable convolution and channel attention mechanism, and uses an accelerator to increase the system's processing efficiency. We implemented the entire system on a custom and low-cost FPGA SOC platform (Xilinx Inc. ZYNQ^{TM} XC7Z035). Experiments can conclude that our approach has compatible performance to GPU-based methods with much lower resource usage.

Keywords: FPGA-based Dehazing · Lightweight Network.

1 Introduction

Images captured by cameras can have poor visibility due to the loss of saturation and contrast caused by the presence of cloudy media such as water vapour, mist, dust and smoke in the atmosphere. With these hazy images as input, autonomous systems such as self-driving cars, intelligent traffic surveillance and unmanned aerial vehicles face degraded performance or severe failures. In addition these systems are used in scenarios where efficiency and low power consumption are sought. Therefore, a dedicated dehazing hardware solution is required to meet these limitations.

Early hardware systems for image dehazing were only designed to speed up software algorithms. Lu et al.[1] combines dark channel prior algorithm to implement an improved fast image dehazing system on a DSP embedded platform. In particular, the system's computing effort is drastically lowered while yet maintaining the highest possible image quality. However, as camera technology continues to evolve and the resolution of images becomes higher and higher, DSP-based hardware systems struggle to keep up with the speed of image processing. To enhance the apparent size of various objects in a depth picture,

Kasauka[2] installs a multi-scale retinex approach using an FPGA. However, when the haze concentration is not uniformly distributed, the image dehazing effect will be unsatisfactory. With the low-power flexibility of FPGAs, Ju et al.[3] successfully dehaze photos using the dark channel prior approach, but processes low-resolution images and still lacks the ability to process high-definition images.

In all, FPGA-based image dehazing system promises for a universal solution to FPGAs with high image defintion. The hardware implementation of deep learning based image dehazing algorithms is constrained by the on-chip memory. Both the image frames and the hardware resources required to implement the dehazing logic need to be stored in on-chip memory. When implementing image dehazing algorithms in hardware, it is a challenging task to perform complex mathematical operations with minimal logic resources without compromising the quality of the output image. In this study, we implemented a convolutional accelerator and a lightweight end-to-end neural network for the FPGA. The processing speed of the system is increased while ensuring the dehazing effect.Lightweight networks can utilise fewer hardware resources. It also requires less consideration when deploying on hardware and enables faster deployment on the hardware side.

2 Related Work

The primary available approaches may be roughly categorized as **prior-based and deep learning-based methods**, with the goal of single picture dehazing being to restore a hazy image to a clear one.

The prior-based technique uses a physical scattering model to produce crisp pictures, but it also needs a natural prior to calculate the transmission map and atmospheric light. He et al.[4] discovered that the majority of partially clear pictures included at least one color channel with multiple extremely low intensity pixels and suggested the Dark Channel Prior (DCP) technique of dehazing.

A non-local prior dehazing technique was suggested by Berman et al.[5] when they noticed that the colors of a clear image can closely resemble hundreds of other colors that group together in small groups in the RGB color space. In order to perform picture dehazing, Zhu et al.[6] suggest using a color attenuation prior approach to learn the scene depth of a hazy image using supervised learning. This method then calculates the transmission and recovers the scene radiance. Prior-based methods only work well when the assumed prior is appropriate, and results are often poor when the prior is not satisfied.

Researchers have started to recognize deep learning-based picture dehazing solutions more and more recently. Although some methods[7–10] have been proposed with high dehazing effect, these models have high model complexity, complex computation and huge storage requirements, which make them difficult to deploy in resource-limited platforms. Ren et al.[11] learn to anticipate the scene transmission map using a coarse-scale neural network, then learn local information that used a fine-scale neural network, and lastly recover a clean picture that use the output of both scales of the network. By learning to derive the

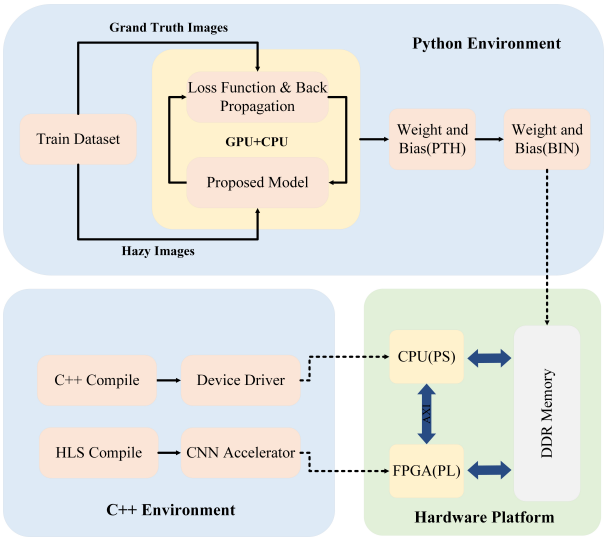

Fig. 1. An overview of the development of the mentioned FPGA-based CNN accelerator for image dehazing.

confidence map of the input, Ren et al.[12] developed a multi-scale gated fusion network based on an encoder decoder network to tackle the single-image dehazing problem. A generative adversarial network (GAN) and an enhancer make up the image dehazing network (DCPDN) that Zhang et al.[13] suggested. The enhancer creates high-quality images after the discriminator directs the generator to create images at a coarse scale.

Although the above mentioned methods are effective in image dehazing, their high number of parameters and high computational effort hinder their deployment on resource-limited platforms. Cai et al.[14] propose the DehazeNet deep learning neural network for estimating media transmission map. The network receives a bad photo as input and then waits for the image to be recovered using its transmission map and atmospheric light. By integrating an atmospheric scattering model into the network and fusing the two variables in the model into a single K parameter, Li et al.[15] design a lightweight trainable end-to-end image dehazing network (AOD-Net) in order to minimize errors. However, the quality of these lightweight networks is not high enough for high resolution image recovery while few people deploy them to hardware platforms, such as FPGAs.

3 Approach

Fig. 1 shows the process of implementing an FPGA-based image dehazing system. First, an embedded-friendly lightweight dehazing deep learning network is designed and trained. Then, a CNN accelerator is designed using the HLS compiler. Finally, a device driver is created using a standard C++ compiler. The

designed and developed accelerator and device driver will control and accelerate the model inference on the hardware platform.

3.1 Network Structure

In this research, we build an encoder-decoder structure-based residual attention-based picture dehazing system. The multi-scale feature extraction blocks, gated fusion sub-network, channel attention block, and encoder-decoder block are the four modules that make up this algorithm's network model. The overall network structure proposed in this paper is shown in Fig. 2. We apply depthwise separable convolution to decrease the total amount of parameters and constructed a lightweight network framework to enable deployment on devices with restricted resources.

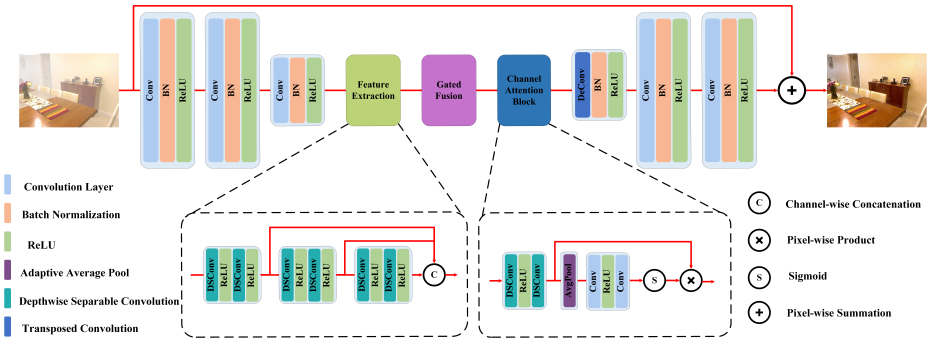

Fig. 2. The overall of our proposed encoder-decoder network.The network contains a feature extraction block, a gated fusion sub-network and a channel attention unit.

Encoder-Decoder Block The input hazy image is first put into the encoder module as a feature map, and the encoder part uses three convolutional layers to learn the haze image, with the last convolutional layer downsampling the feature map by a factor of 1/2. On the contrary, the decoder module contains a transpose convolution to upsample the feature map to it's own original resolution. The next two convolutional layers then nonlinearly map the upsampled feature to produce the desired final hazy residual map.

Feature Extraction Block Spatially separable convolution was used to improve computational efficiency as early as 2012[16]. Sooner or later, a depth-wise version of AlexNet[17] has been added in order to increase accuracy, speed up convergence, and compact the model. Recently, several light-weight network architectures with accuracy, MobileNet[18] and ShuffleNet[19], have been developed for edge devices. Due to the limited computational resources of the FPGA platform, it was found from these research concepts that depth-wise separable convolution could be less complex in terms of computational resources and we introduced it as the base module for feature extraction. Although minimal

parameters are used, the quality of image dehairing is still guaranteed. Each feature extraction block (FEB_i, i = 1, 2, 3) contains two depth-wise separable convolution(DSConv) and two relu layers(ReLU). The feature extraction module $FEB_i(x)$ is represented as

$$FEB_i(x) = ReLU_{i2}(DSC_{i2}(ReLU_{i1}(DSC_{i1}(F_{i-1}(x))))) \tag{1}$$

where $F_{i-1}(x)$ denotes the current input feature and $DSC_i(x)$ denotes the depth-wise separable convolution.

Gated Fusion Sub-network Based on Chen's[20] research, we fuse the characteristics among several layers using a gated fusion sub-network. In order to fuse the feature maps, F_0, F_1 and F_2 are first extracted from the feature extraction block and then linked in series by channel. The fused feature maps are then sent into the gated fusion sub-network. The weights of the preceding three related feature maps (W_0, W_1 and W_2) are the output of the gated fusion sub-network. The last step is to multiply the three relevant feature maps F_0, F_1 and F_2 by the appropriate weight layers. As shown, the gated fusion sub-network is as follows:

$$\begin{gathered}(W_0, W_1, W_2) = Gat\left(F_0, F_1, F_2\right) \\ F_o = W_0 \otimes F_0 + W_1 \otimes F_1 + W_2 \otimes F_2\end{gathered} \tag{2}$$

The CAU receives additional input from the combined feature map F_o. The gated fusion sub-network in this study has three output channels and one kernel size 3x3 convolutional layer with a cascade of F_0, F_1 and F_2 inputs.

Channel Attention Unit Inspired by PCNet[21] with high effective channel attention units, we use the CAU as our basic block in the proposed network. Depth-wise separable convolution, which performs similarly to regular convolution while being more computationally more efficient, is employed to design CAU in order to further minimize the number of parameters. The depth-wise separable convolution is immediately followed by a global average pooling without changing the dimensionality. Then two convolutional layers of 1×1 size for cross-channel information interaction. The weights are then used to adjust the input feature map to produce the output feature map after the feature map has been through the Sigmoid function to obtain the weight values. By weighing and filtering out the prominent characteristics at the present scale instead of the original features for backward propagation, an efficient channel attention is employed to increase the network's efficiency and performance. The efficient channel attention mechanism $CAU_i(x)$ is expressed as

$$\begin{gathered}DSC_i(x) = DSConv_{2i}(ReLU(DSConv_{1i}(F_{i-1})) \\ CAU_i(x) = \sigma(Conv_{2i}(Conv_{1i}(g(DSC_i(x))))) \otimes DSC_i(x)\end{gathered} \tag{3}$$

where $DSConv(x)$ denotes the depth-wise separable convolution, σ denotes the Sigmoid function and $g(x)$ denotes the global average pooling function. By dynamically adjusting the feature map channel weights to reduce redundancy and learning rich contextual information to enhance the network's ability to extract haze density images, the effective Channel Attention Unit (CAU), when used after the gated fusion sub-network, enables a more detailed dehazing.

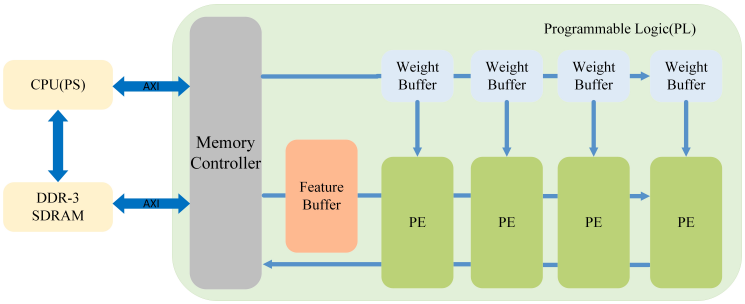

Fig. 3. Hardware architecture of the proposed CNN accelerator for image dehazing.

3.2 Hardware Optimizations

The most frequent and largest operation in the network is convolution operation, because convolution operation requires a lot of multiplication and addition operations, and a large number of access operations on parameters during the operation also consumes a lot of time. The hardware architecture of the dehazing network is designed around the convolution operations and access features in the network layer, so we need to design a convolution accelerator to ensure the speed of convolutional operations while balancing the relationship between hardware resources and memory bandwidth.

Huge data must be stored on off-chip resources due to the FPGA's restricted on-chip capacity. Nevertheless, data from the off-chip memory must first be transmitted through AXI to the on-chip memory within the FPGA before the device can run a parallel on-chip program. Every time the FPGA runs a task, it will have to read data from off-chip memory, which will take a lot of time and cause performance to suffer. We store as much data as we can in FPGA on-chip memory on our limited on-chip resources to prevent constantly reading and writing data from off-chip memory. In designing the memory access section, we used a cyclic partitioning approach. Suppose the size of input feature map In is $N \times H_{in} \times W_{in}$, the weight is $M \times N \times K \times K$, the size of output feature map Out is $M \times H_{out} \times W_{out}$, and the input channel, output channel and output feature map height and width of the partitioning factor are Tn, Tm, Tr and Tc respectively. During each calculation, we load $Tn \times (Tr + K - P) \times (Tc + K - P)$ size of the input feature block, $Tm \times Tn \times K \times K$ weights, and then perform a convolution calculation to obtain an output feature block of size $Tm \times Tr \times Tc$. Once the computation is complete, we will read the input features and weights for the next block and continue the computation until the convolution is complete. As shown in the Fig. 3, we store the intermediate output of each operation on a feature buffer in on-chip memory. This saves a lot of time by reading the data directly from the on-chip each time a new convolution calculation is performed.

The computation of the convolutional layers is that of a multi-layer loop nested, which means that the computation of the convolution is very slow. The PIPELINE command, which defines the loop to be enlarged and informs the

compiler how many times the loop needs to be expanded, is used to optimize the program's parallel execution speed. By running computations in parallel and quickening the system's inference, PIPELINE takes use of the parallelism between the convolution kernels to maximize the usage of processing resources.

4 Results and Analysis

4.1 Experimental setup and data set

We refer to the work of [20] for network architecture design and training. An Intel(R) Xeon(R) Processor E5-2620 v3 @ 2.40GHz processor, 16.0GB of Memory, and two NVIDIA Titan Xp graphics cards made up the experimental setup. The network was created using the Pytorch framework, with a training batch size of 16. The learning rate started out at 0.01 and declined to 0.1 times every 40 iterations for a total of 100. We used the Indoor Training Set (ITS), a subset of the publicly accessible image dehazing RESIDE[22] dataset, as the training dataset. Using a special development board made by Xillinx Inc., the suggested image dehazing technology is put into practice. It was made up of an XC7Z035 FPGA and a dual-core ARM Cortex-A9 CPU.Prior to deployment, we performed a tuning optimization of the network. There are a large number of convolutional and batch normalisation layer structures in the network. Due to their computational properties, we can reduce the model inference time by merging the batch normalization layers into the convolutional layers.

4.2 Comparisons with state-of-arts

Compare with Other GPU-Based Methods Table 1 gives the results of this experiment tested on synthetic datasets and compared quantitatively and qualitatively with recent methods, including AODNet[15], DehazeNet[14], DCPDN[13] and GCANet[20]. Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) index measurements were also used for quantitative evaluation, with higher values indicating greater dehazing. Our approach gives a competitive performance to GPU-based methods with a relative small parameter size, which is around 0.1 million. For visual comparsions on the SOTS test set, hazy pictures of various intensities were also chosen for the evaluation of subjective quality. Fig. 4 displays the dehazing impacts of each approach together with the related peak signal-to-noise ratios. The photos that were recovered using AODNet techniques had dark colors and insufficient dehazing. Nevertheless, when eliminating intense haze, the pictures recovered by DehazeNet and DCPDN approaches are prone to insufficient dehazing. Our approach shows a better dehazing effect, no obvious color distortion, more complete dehazing of dense hazy images, and the recovered image details and colors are closer to the original clear image. Although the dehazing effect of GCANet is slightly better than the method in this paper, the model size of GCANet is about 7 times larger than ours.And when processing some photographs with sections of sky, GCANet warps the colors. By contrast, our method does a good job of restoring the original colours.

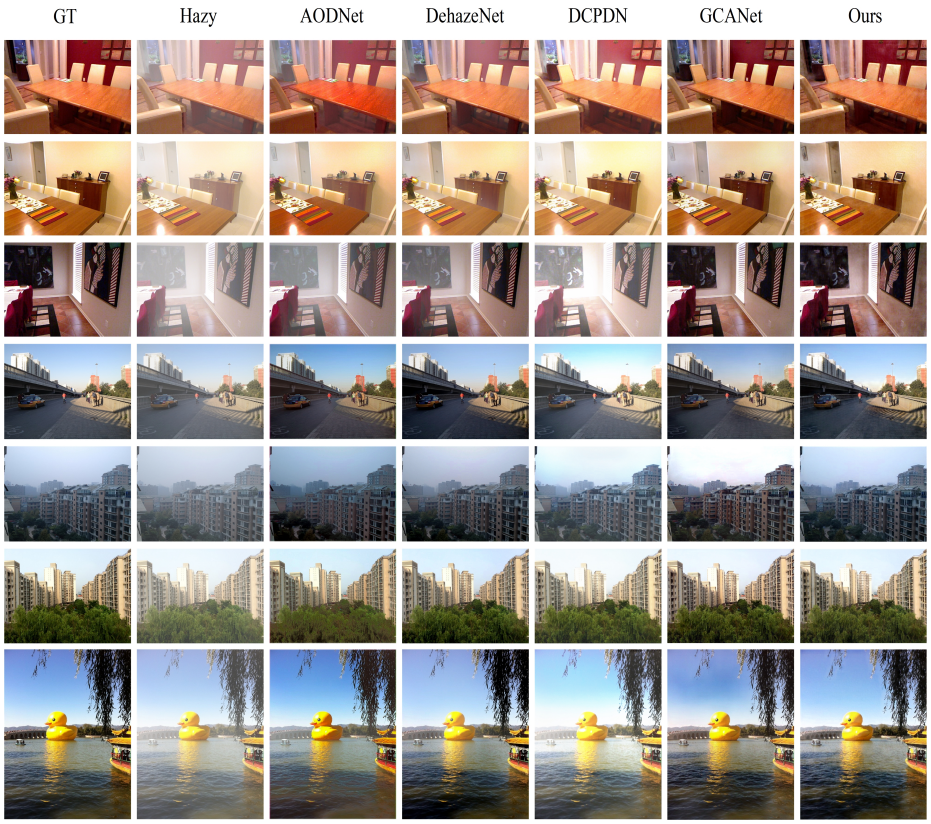

Fig. 4. Qualitative comparison of different image dehazing methods on the SOTS dataset.

Compare with Other Traditional Methods As for the evaluation dataset, as deep learning based methods are mostly trained on the RESIDE dataset, their performance on this dataset is better than other datasets. For the sake of fairness, when comparing with traditional methods, we employ FRIDA2[24], D-HAZY[25], O-HAZE[26], I-HAZE[27], and Dense-Haze[28]. Moreover, we employ Mean Square Error (MSE) and Structural Similarity (SSIM), two full-reference criteria, to quantitatively assess the dehazing performance. The results of this experiment tested on the five datasets mentioned previously and compared with conventional FPGA-based dehazing methods, those proposed by He et al.[4] , Zhu et al.[6] , Berman et al.[5] , and Cho et al.[23], are given in the Table 2. The traditional methods cannot effectively handle the sky region leading to their poor performance on FRIDA2. On the contrary, our approach ranks first in the overall SSIM evaluation, which is the primary indicator for visble edges, on five databases from different conditions.

Table 1. Quantitative comparisons of image dehazing on the SOTS indoor dataset from RESIDE.

Methods	PSNR	SSIM	Par. (Million)
AODNet[15]	19.06	0.85	0.002
DehazeNet[14]	21.14	0.84	0.0802
DCPDN[13]	15.85	0.82	66.89
GCANET[20]	30.06	0.96	0.7028
Ours	27.26	0.93	0.1297

Table 2. Scores for structural similarity, peak signal to noise ratio (PSNR), and mean squared error (MSE) on various datasets. The best result is shown in red.

	He et al.[4]		Zhu et al.[6]		Berman et al.[5]		Cho et al.[23]		Our	
DATASET	MSE	SSIM	MSE	SSIM	MSE	SSIM	MSE	SSIM	MSE	SSIM
FRIDA2	0.0744	0.5969	0.0744	0.5473	0.0705	0.6603	0.1559	0.5517	0.0642	0.6687
D-HAZY	0.0309	0.8348	0.0483	0.7984	0.0492	0.7473	0.0606	0.7212	0.0458	0.7629
O-HAZE	0.0200	0.7709	0.0226	0.6647	0.0255	0.8024	0.0196	0.7745	0.0197	0.6319
I-HAZE	0.0535	0.6580	0.0362	0.6864	0.0275	0.7959	0.0344	0.7693	0.0308	0.7169
Dense-Haze	0.0549	0.4662	0.0646	0.4171	0.0597	0.5225	0.0549	0.5254	0.0613	0.4174
Total	0.0467	0.6653	0.0492	0.6227	0.0464	0.7056	0.0650	0.6684	0.0444	0.6396

4.3 Ablation Study

We combine the depth-wise separable convolution, channel attention unit, and gated fusion sub-networks into our suggested model, as was already indicated. To confirm the contribution of each component to the final dehazing performance, we conduct ablation experiments on the SOTS dataset.Instead of using depth-wise separable convolution, we use conventional convolution instead, and we change the number of convolutions to roughly equalize the size of the network overall.As demonstrated in the Table 3, the introduction of the channel attention module significantly improves the model's performance, demonstrating the efficacy of CAU. Subsequently, the performance of the model is somewhat improved by the addition of deep separable convolution and gated sub-networks. Eventually, including all three modules into the model yields the optimal outcome.

4.4 Hardware Evaluation

In this subsection, we first provide the resource utilization rate. Then, we compared the software implementation (on CPU and GPU) with our accelerator on FPGA. Placement and wiring are done through the Vivado toolset. After the collection is completed, the resource utilization rate we achieved is reported, as shown in the Table 4. We can see that our CNN accelerator requires very little FPGA hardware resources.We compared our method on FPGA with other platforms. We selected NVIDIA Titan-Xp GPU and Intel i7-8700 CPU for comparison. We tested the power consumption of the FPGA end, and the power

Table 3. Ablation study on SOTS dataset.

Depth-Wise Convolution		✓			✓
Gated Fusion			✓		✓
CAU				✓	✓
PSNR	22.93	24.24	23.43	26.44	27.26
Par. (Million)	0.1288	0.1250	0.1309	0.1336	0.1297

Table 4. Convolutional Accelerator Resource utilization.

Resource	DSP48E	BRAM_18K	LUT	FF
Used	213	161	60803	58930
Available	900	1000	171900	343800
Utilization	23	16	35	17

consumption of the GPU and CPU came from the user manual. From the Table 5, it can be concluded that our method is suitable for edge platforms with low power consumption and few resources.

Table 5. Comparison with other platforms.

Platform	GPU	CPU	FPGA
Device	Titan Xp	i7-8700	XC7Z035
Power(Watt)	250	85	4.2
Performance(FPS)	394	1.5	8.3
Energy efficiency(FPS/W)	1.57	0.017	1.97

5 Conclusion

In this research, we provide a lightweight FPGA-based deep learning-based approach for image dehazing. A CNN network based on depth-wise separable convolution and channel attention to limit the network size is suggested in order to lower the storage and computing requirements.Then, using our suggested accelerated design strategy, we deploy the entire algorithm on a low-cost custom FPGA development board from Xilinx Inc. Therefore, our method is an universal solution to image dehazing on FPGAs.

Acknowledgments

This work was supported by the Natural Science Foundation of China (62202347) and the Natural Science Foundation of Hubei Province (2022CFB578).

References

1. Jinzheng Lu and Chuan Dong. Dsp-based image real-time dehazing optimization for improved dark-channel prior algorithm. *Journal of Real-Time Image Processing*, 17(5):1675–1684, 2020.
2. Dabwitso Kasauka, Kenta Sugiyama, Hiroshi Tsutsui, Hiroyuki Okuhata, and Yoshikazu Miyanaga. An architecture for real-time retinex-based image enhancement and haze removal and its fpga implementation. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 102(6):775–782, 2019.
3. Y. K. Ju and Jae Wook Jeon. Implementation of a single-image haze removal using the fpga. In *the 12th International Conference*, 2018.
4. Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2341–2353, 2010.
5. Dana Berman, Shai Avidan, et al. Non-local image dehazing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1674–1682, 2016.
6. Qingsong Zhu, Jiaming Mai, and Ling Shao. A fast single image haze removal algorithm using color attenuation prior. *IEEE transactions on image processing*, 24(11):3522–3533, 2015.
7. Chun-Le Guo, Qixin Yan, Saeed Anwar, Runmin Cong, Wenqi Ren, and Chongyi Li. Image dehazing transformer with transmission-aware 3d position embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5812–5820, 2022.
8. Tian Ye, Mingchao Jiang, Yunchen Zhang, Liang Chen, Erkang Chen, Pen Chen, and Zhiyong Lu. Perceiving and modeling density is all you need for image dehazing. *arXiv preprint arXiv:2111.09733*, 2021.
9. Yu Zheng, Jiahui Zhan, Shengfeng He, Junyu Dong, and Yong Du. Curricular contrastive regularization for physics-aware single image dehazing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.
10. Yuda Song, Yang Zhou, Hui Qian, and Xin Du. Rethinking performance gains in image dehazing networks. *arXiv preprint arXiv:2209.11448*, 2022.
11. Wenqi Ren, Si Liu, Hua Zhang, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. Single image dehazing via multi-scale convolutional neural networks. In *European conference on computer vision*, pages 154–169. Springer, 2016.
12. Wenqi Ren, Lin Ma, Jiawei Zhang, Jinshan Pan, Xiaochun Cao, Wei Liu, and Ming-Hsuan Yang. Gated fusion network for single image dehazing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3253–3261, 2018.
13. He Zhang and Vishal M Patel. Densely connected pyramid dehazing network. In *CVPR*, 2018.
14. Bolun Cai, Xiangmin Xu, Kui Jia, Chunmei Qing, and Dacheng Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, 2016.
15. Boyi Li, Xiulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. Aod-net: All-in-one dehazing network. In *Proceedings of the IEEE international conference on computer vision*, pages 4770–4778, 2017.

16. Franck Mamalet and Christophe Garcia. Simplifying convnets for fast learning. In *Artificial Neural Networks and Machine Learning–ICANN 2012: 22nd International Conference on Artificial Neural Networks, Lausanne, Switzerland, September 11-14, 2012, Proceedings, Part II 22*, pages 58–65, 2012.
17. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.
18. Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.
19. Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.
20. Dongdong Chen, Mingming He, Qingnan Fan, Jing Liao, Liheng Zhang, Dongdong Hou, Lu Yuan, and Gang Hua. Gated context aggregation network for image dehazing and deraining. In *2019 IEEE winter conference on applications of computer vision (WACV)*, pages 1375–1383. IEEE, 2019.
21. K. Jiang, Z. Wang, P. Yi, C. Chen, and C. W. Lin. Rain-free and residue hand-in-hand: A progressive coupled network for real-time image deraining. *IEEE Transactions on Image Processing*, 2021.
22. Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE Transactions on Image Processing*, 28(1):492–505, 2018.
23. Younggun Cho, Jinyong Jeong, and Ayoung Kim. Model-assisted multiband fusion for single image enhancement and applications to robot vision. *IEEE Robotics and Automation Letters*, 3(4):2822–2829, 2018.
24. Jean-Philippe Tarel, Nicholas Hautiere, Laurent Caraffa, Aurélien Cord, Houssam Halmaoui, and Dominique Gruyer. Vision enhancement in homogeneous and heterogeneous fog. *IEEE Intelligent Transportation Systems Magazine*, 4(2):6–20, 2012.
25. Cosmin Ancuti, Codruta O Ancuti, and Christophe De Vleeschouwer. D-hazy: A dataset to evaluate quantitatively dehazing algorithms. In *2016 IEEE international conference on image processing (ICIP)*, pages 2226–2230. IEEE, 2016.
26. Codruta O Ancuti, Cosmin Ancuti, Radu Timofte, and Christophe De Vleeschouwer. O-haze: a dehazing benchmark with real hazy and haze-free outdoor images. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 754–762, 2018.
27. Cosmin Ancuti, Codruta O Ancuti, Radu Timofte, and Christophe De Vleeschouwer. I-haze: a dehazing benchmark with real hazy and haze-free indoor images. In *Advanced Concepts for Intelligent Vision Systems: 19th International Conference, ACIVS 2018, Poitiers, France, September 24–27, 2018, Proceedings 19*, pages 620–631. Springer, 2018.
28. Codruta O Ancuti, Cosmin Ancuti, Mateu Sbert, and Radu Timofte. Dense-haze: A benchmark for image dehazing with dense-haze and haze-free images. In *2019 IEEE international conference on image processing (ICIP)*, pages 1014–1018. IEEE, 2019.