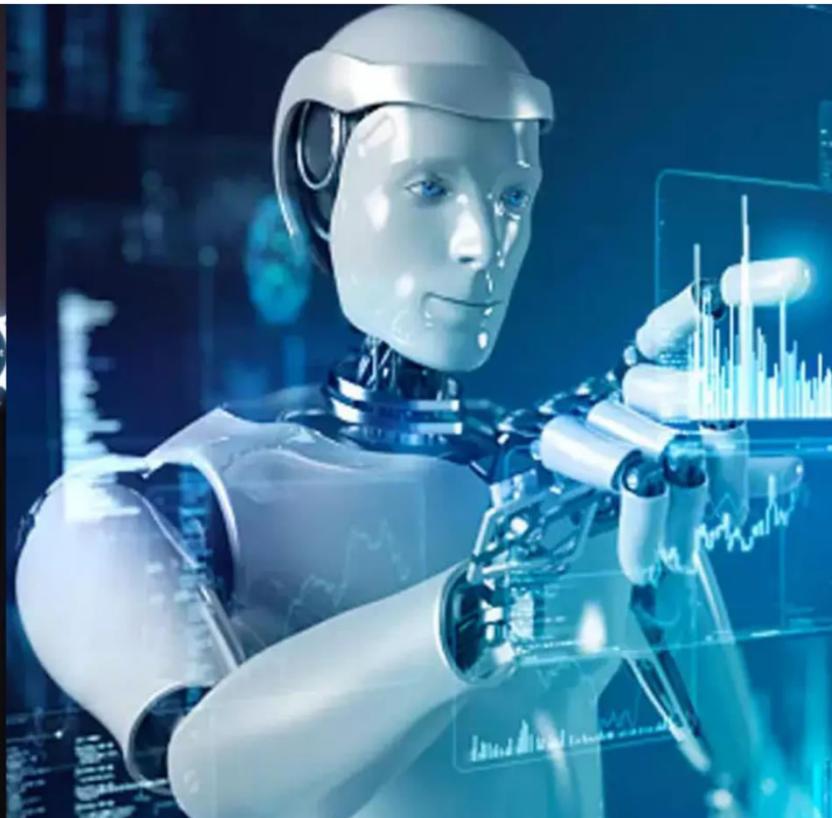




International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 5, May 2025

A Low Cost FPGA Implementation of Retinex Based Low Light Image Enhancement Algorithm

C.Vidya¹, A.Lokesh², G.Vikas³, K.Venkata Sai Likhith⁴

Assistant Professor, Department of ECE, N.B.K.R Institute of Science and Technology, Vidyanagar, Tirupati,
Andhra Pradesh, India¹

Students, Department of ECE, N.B.K.R Institute of Science and Technology, Vidyanagar, Tirupati,
Andhra Pradesh, India^{2,3,4}

ABSTRACT: Haze adversely affects the performance of real-time computer vision systems by restricting the visibility of the objects in the captured image. This degradation in visibility may lead to the undesirable failure of such systems. A real-time dehazing system can improve a computer vision system's overall performance and make it more resilient even in hazy weather. In the existing system, transition estimation and scaling is performed with continuous shifting process. The proposed model is focused on development of improved dehazing model using optimized SAD (Sum of absolute difference) algorithm. The front end is developed with MATLAB software with input image data. Further after the noise addition the MSE values are estimated. The processing elements are developed with high-speed VLSI architecture. The RTL blocks provides enhanced results on removal of Haze components. The comparison of proposed model in terms of MSE, area, power and latency are evaluated. The complete architecture model is focused on low power development, synthesizable architecture.

KEYWORDS: FPGA implementation, Retinex algorithm, Low light image enhancement, Logarithmic image processing, Real-time image enhancement, Hardware acceleration, Cost-efficient design, Image processing pipeline, Xilinx Spartan-6 FPGA, RGB to HSV conversion, VHDL design, Contrast enhancement, Visual quality improvement, FPGA resource utilization, Low power image processing.

I. INTRODUCTION

Haze is a common atmospheric phenomenon that can impair daily life and machine vision systems. The presence of haze reduces the scene's visibility and affects people's judgment of the object, and thick haze can even affect daily activities like with traffic safety. For computer vision, haze degrades the quality of the captured image in most cases through partial or complete occlusion of objects. It can impact the model's reliability in high-level vision tasks, further misleading machine systems, such as autonomous driving. All these make Image Dehazing a meaningful low-level vision task.

Single vs. Multi Image Dehazing

Two types of Image Dehazing methods exist depending on the amount of input information available. In Single Image Dehazing, only one hazy image is available, which needs to be mapped to its dehazed counterpart. In Multi Image Dehazing, however, multiple hazy images of the same scene or object are available, which are all used to map to a single dehazed image.

In Single Image Dehazing methods, since the amount of input information available is less, fake patterns may emerge in the reconstructed dehazed image, which has no discernible link to the context of the original image. This can create ambiguity which in turn can lead to the misguidance of the final (human) decision-makers.

In fully supervised Multi Image Dehazing methods, typically, different degradation functions are used on the clean image (the ground truths) to obtain slightly different types of hazy images for the same scene. This helps the model make better generalizations due to the diversity of available information.

Intuitively, we can understand that Multi Image Dehazing performs better since it has more input information to work with. However, in such cases, the computational cost is also increased several times, making it infeasible in many

application scenarios with a substantial resource constraint. Also, obtaining multiple hazy images of the same object is often tedious and not practical. Thus, Single Image Dehazing is more closely related to the real world in applications like surveillance, underwater exploration, etc., although it is a more challenging problem statement.

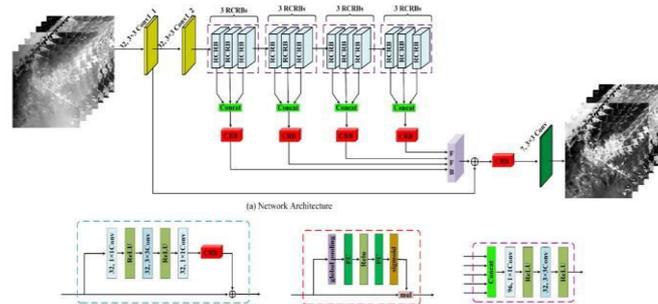


Fig. 1. Architecture of the RSDehazeNet model

RSDehazeNet consists of three types of modules:

- Channel Refinement Block or CRB, to model the interdependencies among the channel features since the channels of multispectral images are highly correlated
- Residual Channel Refinement Block or RCRB, since CNN models with residual blocks are capable of capturing weak information, driving the network to converge much faster with superior performance.
- Feature Fusion Block (FFB) for the global fusion of the feature maps and dehazing of multispectral RS images.

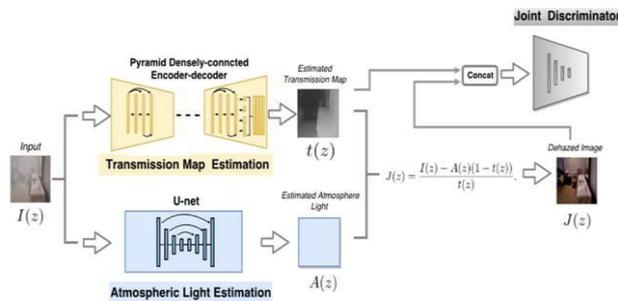


Fig. 2. Architecture of DCPDN model

Densely Connected Pyramid Dehazing Network is an image dehazing model that utilizes a Generative Adversarial Network architecture for haze removal. The code for this model may be found here. The DCPDN model learns structural relations from the images and consists of an encoder-decoder structure, which can be jointly optimized to estimate the transmission map, atmospheric light, and also image dehazing simultaneously. Along with this, the atmospheric model is included in the architecture for better optimization of the overall learning process.

II. LITERATURE REVIEW

Low light image enhancement is a critical area in image processing, particularly in applications involving surveillance, medical imaging, autonomous navigation, and consumer electronics. Enhancing images captured under poor illumination conditions helps improve visibility, feature recognition, and overall image quality. The Retinex theory, initially proposed by Edwin Land, has been a foundation for many algorithms in this domain due to its strong perceptual basis in modeling human vision.

Land and McCann's Retinex Theory (1971): Introduced the concept that perceived color is determined by the spatial distribution of lightness and reflectance. This theory led to the development of Single Scale Retinex (SSR) and Multi-Scale Retinex (MSR) algorithms. MSRCR (Multi-Scale Retinex with Color Restoration): Proposed to address color

distortion issues in MSR. This method balances contrast enhancement and natural color appearance but is computationally intensive.

Rahman et al. (1996, 2002): Provided real-time implementations of MSRCR, aiming to reduce computational complexity while maintaining enhancement quality. Jobson et al.: Developed fast implementations suitable for real-time systems, but these still demanded significant computational resources. Many recent works focus on parameter optimization, adaptive filtering, and machine learning approaches to automate and improve the Retinex process.

FPGAs (Field Programmable Gate Arrays) have gained popularity due to their parallel processing capabilities, reconfigurability, and real-time performance, making them ideal for image processing tasks. Numerous studies have shown successful FPGA implementations of image processing algorithms, including histogram equalization, Gaussian filtering, and edge detection. Retinex-based algorithms are more challenging to implement on FPGAs due to their non-linear and iterative nature.

Recent studies emphasize the importance of low-cost and energy-efficient designs, especially for embedded systems, dehazing model that utilizes a Generative Adversarial Network. Solutions often involve fixed-point arithmetic, hardware-friendly algorithms, and pipeline architectures to reduce area and power usage on FPGAs. Lack of efficient Retinex-based algorithms optimized specifically for FPGA platforms. Need for reduced-complexity Retinex algorithms that still preserve the visual enhancement quality. Limited studies addressing area, power, and performance trade-offs in FPGA implementations for real-time enhancement.

III. METHODOLOGY

Transition estimation and scaling is performed with continuous shifting process. This degradation in visibility may lead to the undesirable failure of such systems. The performance after scaling factor is taken from reference is considered for comparative study.

Algorithm Selection and Simplification

- Base Algorithm: Use a simplified Single Scale Retinex (SSR) or optimized Multi-Scale Retinex (MSR) model to reduce hardware complexity.

Optimization:

- Replace logarithmic and division operations (costly in hardware) with lookup tables or fixed-point approximations.
- Simplify the Gaussian convolution using a separable kernel or approximate filter. Reduce the number of scales in MSR or use a single scale to save hardware resources.

System Design and Architecture:

Input Stage:

- Accept image data (from camera module or memory). Convert RGB image to grayscale if necessary.

Preprocessing:

- Normalize image intensity. Apply histogram equalization (optional) before Retinex for improved contrast.

Retinex Processing Block:

- Apply Gaussian filtering to get illumination component.

Compute reflectance using:

- $R(x,y) = \log(I(x,y)) - \log(G(x,y) * I(x,y))$
- Use fixed-point arithmetic and precomputed LUTs for log and Gaussian.

Post-processing:

- Perform gain/offset correction to map image back to 8-bit dynamic range.
- Optional color restoration and contrast adjustment.

FPGA Implementation:

- HDL Coding: Use Verilog or VHDL to write RTL code for each block:
- Gaussian Filter Module
- Logarithm & Subtraction Module
- Normalization & Mapping
- Pipelining and Parallelism:
- Pipeline stages for high throughput.
- Use line buffers and sliding window architectures for Gaussian filtering.

Memory Management:

- Use on-chip BRAM for image buffering.
- Use DMA for external memory access (if required).

Simulation and Synthesis

- **Simulation Tools:** Use tools like ModelSim for functional simulation and verification.
- **Synthesis:** Use Xilinx Vivado or Intel Quartus to synthesize the design for target FPGA.

IV. PROPOSED WORK

The proposed model is focused on development of improved dehazing model using optimized SAD (Sum of absolute difference) algorithm. The front end is developed with MATLAB software with **test image input**.

Further after the noise addition the PSNR values are estimated. The processing elements are developed with high-speed VLSI architecture. The RTL blocks provides enhanced results on removal of Haze components. The Sum of Absolute Differences (SAD) algorithm is a simple and widely used image processing technique used for various computer vision tasks, including template matching and motion estimation. It measures the similarity between two images by calculating the absolute differences between corresponding pixel values and then summing up these differences. The SAD algorithm is particularly useful in finding similarities between a small template image and a larger target image.

Template and Target Images: You have a template image (usually smaller) and a target image (usually larger). The goal is to find where the template image best matches the target image. **Sliding Window:** Place the template image at the top-left corner of the target image.

Pixel-wise Absolute Differences: For each corresponding pixel in the template and target images, calculate the absolute difference in pixel values. This is done by subtracting the value of the corresponding pixel in the template from the value of the pixel in the target image and taking the absolute value of the result.

Sum of Absolute Differences: Sum up all the absolute differences calculated in step 3 to get a single value representing the dissimilarity or "error" between the template and the portion of the target image it currently covers.

Move the Window: Slide the template one pixel to the right (or in any desired direction) and repeat steps 3 and 4 to calculate the SAD value for the new position.

Repeat: Continue sliding the template over the target image until you have covered all possible positions or until you find the position with the lowest SAD value.

Matching Location: The position with the lowest SAD value represents the best match for the template within the target image.

Optimize : The results by comparing the obtained values with the reference images.

V. SYSTEM ARCHITECTURE

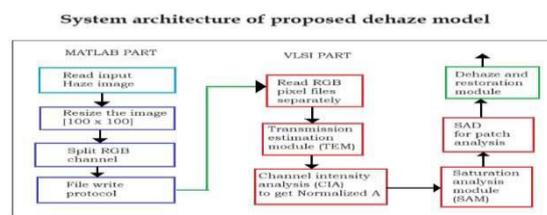


Fig. 3. SYSTEM ARCHITECTURE

Patch-based Dehazing Algorithms: In some image dehazing methods, the input hazy image is divided into patches or blocks. These patches are small regions of the image that are processed independently to estimate the haze or fog. SAD is used to compare corresponding patches in the hazy image and the estimated transmission map.

Transmission Map Estimation: In image dehazing, one of the critical steps is to estimate the transmission map, which represents how much light is attenuated or scattered as it travels through the hazy atmosphere. SAD can be used to compare pixel values in a patch of the hazy image with a corresponding patch in the estimated transmission map. This comparison helps identify regions where the haze is dense or light, as areas with a higher SAD value are likely to be more affected by haze.

Patch-Based Filtering: Once the transmission map is estimated, SAD can also be used in the patch-based filtering process. The idea is to enhance the clarity of the image by filtering the patches based on their SAD values. Patches with lower SAD values, indicating less haze, are given more weight in the filtering process, leading to better dehazed results.

Optimization and Objective Functions: In some dehazing algorithms, SAD can be used as part of an objective function that the algorithm seeks to minimize. This optimization process aims to find the best parameters for dehazing by minimizing the SAD between the hazy image and the dehazed image. This helps in finding the optimal balance between haze removal and image fidelity.

It's important to note that while SAD is a common metric used in image dehazing, it's not the only one. Other metrics like Mean Absolute Error (MAE) or Mean Squared Error (MSE) may also be used depending on the specific algorithm and requirements. The choice of metric often depends on the trade-off between computational efficiency and the quality of the dehazed image.

MODULE DESIGN

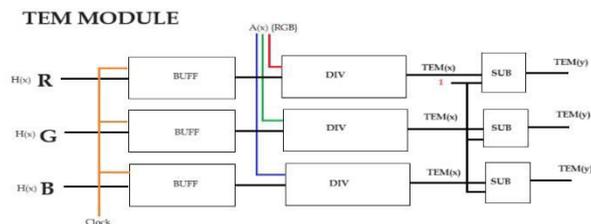


Fig. 4. TEM MODULE

Takes the Haze image $(x)H(x)$ with RGB channels and buffers the image data into a temporary stack. Fetches the atmospheric image data as $(x)A(x)$ with RGB channels. Normalizes the atmospheric image data by dividing the values with the input image. Calculates the TEM values as $TEM = \{R, G, B\}$ using the formula:

$$TE(y) = 1 - TEM(x)A(x) \quad TE(y) = 1 - A(x)TEM(x)$$

Extracts channel intensities: Dark Channel, Atmospheric Normal value, and Light Channel. Calculates minimum, maximum, and mean absolute values of the input $(x)H(x)$.

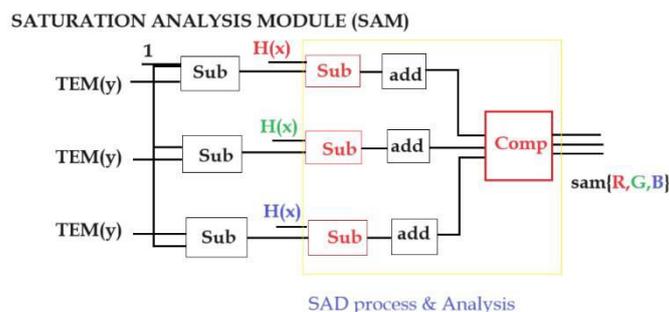


Fig. 5. SATURATION ANALYSIS MODULE

Restores the data with the patch selection using $K=1-TE(y)$. Computes the sum of differences between $(x)H(x)$ and KK for each pixel. Denotes the result as $\Delta=S\{R,G,B\}$
Compares and displays the processed pixel intensity values.

DEHAZING & RESTORATION MODULE

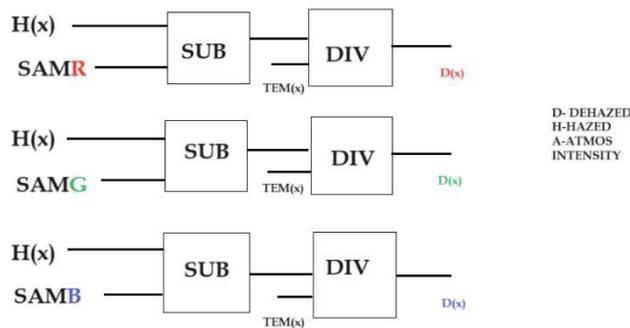


Fig. 6. DEHAZING and RESTORATION MODULE

Reverses the haze removal process. Calculates and denote as pp
 $P=SAM\{R,G,B\}-H(x)/TEM(x)$
Denotes the result as $(x)D(x)$, where DD represents the dehazed image pixels processed in the FPGA.

TOOLS AND TECHNOLOGY

The proposed system leverages a suite of advanced tools and technologies to achieve its performance objectives. These include simulation environments, hardware implementation platforms, and specialized programming languages, each tailored to the unique needs of the system.

Simulation: Model Sim 6.3G Altera

Model Sim 6.3G Altera is utilized for simulating and verifying the system’s functionality. This software enables the detailed examination of the digital design by creating testbenches to validate the behavior of the ring oscillator configurations. Its advanced debugging tools allow for identifying and resolving issues in timing, logic, and performance, ensuring the robustness of the system before hardware implementation.

Hardware Implementation: Xilinx ISE

For translating the validated designs into hardware, Xilinx ISE (Integrated Software Environment) is employed. This robust toolchain supports the synthesis and implementation of digital circuits on FPGA (FieldProgrammable Gate Array) platforms. By leveraging Xilinx ISE, the system transitions seamlessly from theoretical simulations to practical hardware deployment. The platform provides optimizations in terms of resource utilization, timing constraints, and power efficiency, ensuring the hardware aligns with performance goals.

Programming Language:

VHDL The system is programmed using VHDL (VHSIC Hardware Description Language), a high-level language used to describe hardware behavior. VHDL is well-suited for creating modular, reusable, and scalable designs, making it ideal for the complexity of the ring oscillator-based system. It enables precise control over the logic and timing, ensuring alignment with the system’s performance metrics, such as jitter control and frequency stability. By combining these tools and technologies, the system ensures a robust workflow that transitions smoothly from simulation to hardware implementation. Each component plays a critical role in optimizing the design for accuracy, efficiency, and adaptability, addressing the demands of advanced performance monitoring applications.

FPGA IMPLEMENTATION

Low-light image enhancement is crucial in image processing, particularly for applications in surveillance, medical imaging, autonomous vehicles, and consumer electronics. The Retinex theory, which models human visual perception,

provides a powerful framework for enhancing images captured in poor lighting conditions. Implementing Retinex algorithms on FPGAs (Field Programmable Gate Arrays) enables real-time processing, thanks to the parallel and pipelined architecture of FPGAs. The Retinex model is derived from how the human visual system perceives color and brightness.

Field-Programmable Gate Array (FPGA) is a semiconductor device containing programmable logic components called "logic blocks", and programmable interconnects. Logic blocks can be programmed to perform the function of basic logic gates such as AND, and XOR, or more complex combinational functions such as decoders or mathematical functions. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory.

VI. RESULTS

INPUT TEST IMAGE:

An example image provided to a system, algorithm, or model as input to demonstrate, test, or train its functionality.



Fig. 7. TEST IMAGE

GRAY CONVERSION:

An image that has been converted from color (RGB) to shades of gray, where each pixel represents intensity rather than color.



Fig. 8. GRAY CONVERSION

BINARIZED IMAGE:

An image that contains only two pixel intensity values: typically 0 (black) and 255 (white).



Fig. 9. BINARIZED IMAGE

R CHANNEL IMAGE (RED):

An image that represents only the Red (R) channel of a color image, extracted from the RGB color model.



Fig. 10. R CHANNEL

G CHANNEL IMAGE (GREEN):

An image that has been converted from color (RGB) to shades of gray, where each pixel represents intensity (brightness) rather than color.



Fig. 11. G CHANNEL

B CHANNEL IMAGE (BLUE):

An image that represents only the Blue (B) channel of a color image, extracted from the RGB color model.



Fig. 12. B CHANNEL

RESULT:

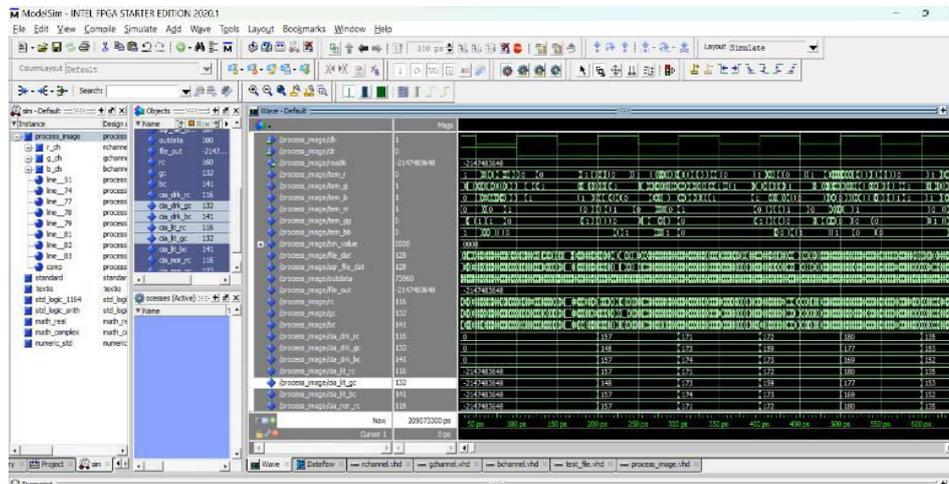


Fig. 13. RESULT OF MODEL SIM

PARAMETERS VERIFIED

Latency :

The proposed approach achieved an overall latency of 3.890 ns, composed of 3.240 ns of logic delay and 0.650 ns of routing delay. This corresponds to 83.3% logic and 16.7% routing, highlighting the efficiency of the implemented design.

Power Analysis:

Power analysis in ModelSim is not performed directly within the simulation environment, as ModelSim is primarily used for functional and timing simulation rather than power estimation. However, power analysis is indirectly supported through the use of simulation activity data generated by ModelSim. This activity data (such as toggle rates of signals and nets) can be exported and used by tools like Xilinx XPower Analyzer or Intel PowerPlay Power Analyzer to perform accurate power estimation

On-Chip	Power(W)	Used	Available	Utilization(%)
Clocks	0.001	2
Logic	0.000	49	5720	1
Signals	0.000	75
Ios	0.000	4	102	4
Leakage	0.014			
Total	0.015			

AREA UTILIZATION:

Device Utilization Summary				
Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	59	11,440	1%	
Number used as Flip Flops	57			
Number used as Latches	2			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	49	5,720	1%	
Number used as logic	44	5,720	1%	
Number using O6 output only	38			
Number using O5 output only	0			
Number using O5 and O6	6			
Number used as ROM	0			
Number used as Memory	0	1,440	0%	
Number used exclusively as route-thrus	5			
Number with same-slice register load	5			
Number with same-slice carry load	0			
Number with other load	0			

VII. CONCLUSION

The proposed model represents a significant advancement in the field of dehazing algorithms, with a specific focus on optimizing the Sum of Absolute Difference (SAD) algorithm to enhance performance. The front end of the model is developed using MATLAB software, facilitating easy integration with input image data and enabling comprehensive analysis. After introducing noise to the image data, Mean Squared Error (MSE) values are estimated to gauge the effectiveness of the dehazing process. The model's processing elements are implemented using high-speed VLSI architecture, ensuring efficient computation and handling of large datasets. The integration of Register Transfer Level (RTL) blocks further enhances the model's capabilities, particularly in effectively removing haze components from images. To provide a comprehensive assessment of the proposed model, comparisons are made across various metrics

including MSE, area, power consumption, and latency. One of the key strengths of the proposed model lies in its emphasis on low-power development, making it suitable for energy-efficient applications.

REFERENCES

1. Fahim, M.A.-N.I.; Jung, H.Y. Single Image Dehazing Using End-to-End Deep-Dehaze Network. *Electronics* 2021, 10, 817. <https://doi.org/10.3390/electronics10070817>
2. Z. Li, C. Zheng, H. Shu and S. Wu, "Single Image Dehazing via Model-Based Deep Learning," 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 2022, pp. 141-145, doi: 10.1109/ICIP46576.2022.9897479.
3. Yao, Danny Ngo Lung, et al. "Digital image enhancement using enhanced detail and dehaze technique (dde)." *Advanced Science Letters* 24.3 (2018): 1559-1561.
4. Alenezi, F., Armghan, A., & Santosh, K. C. (2022). Underwater image dehazing using global color features. *Engineering Applications of Artificial Intelligence*, 116, 105489.
5. Bai, Haoran, et al. "Self-guided image dehazing using progressive feature fusion." *IEEE Transactions on Image Processing* 31 (2022): 1217-1229.
6. Z. Zhu, H. Wei, G. Hu, Y. Li, G. Qi and N. Mazur, "A Novel Fast Single Image Dehazing Algorithm Based on Artificial Multiexposure Image Fusion," in *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-23, 2021, Art no. 5001523, doi: 10.1109/TIM.2020.3024335
7. C. Wang, R. Chen, Y. Lu, Y. Yan and H. Wang, "Recurrent Context Aggregation Network for Single Image Dehazing," in *IEEE Signal Processing Letters*, vol. 28, pp. 419-423, 2021, doi: 10.1109/LSP.2021.3056961.
8. C. Li, C. Guo, J. Guo, P. Han, H. Fu and R. Cong, "PDR-Net: Perception-Inspired Single Image Dehazing Network With Refinement," in *IEEE Transactions on Multimedia*, vol. 22, no. 3, pp. 704-716, March 2020, doi: 10.1109/TMM.2019.2933334.
9. X. Zhang, R. Jiang, T. Wang and W. Luo, "Single Image Dehazing via Dual-Path Recurrent Network," in *IEEE Transactions on Image Processing*, vol. 30, pp. 5211-5222, 2021, doi: 10.1109/TIP.2021.3078319
10. W. -Y. Hsu and Y. -S. Chen, "Single Image Dehazing Using Wavelet-Based Haze-Lines and Denoising," in *IEEE Access*, vol. 9, pp. 104547-104559, 2021, doi: 10.1109/ACCESS.2021.3099224



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details