Jurnal E-Komtek
Vol. 9, No. 1. (2025) pp. 182-197
https://jurnal.politeknik-kebumen.ac.id/index.php/E-KOMTEK
p-ISSN : 2580-3719 e-ISSN : 2622-3066

Single Image Dehazing using U-Net and Lightweight Vision Transformer

Dion Aditya[1], Efy Yosrita

[1,2] Department of Computer Science, Institut Teknolog PLN, Indonesia, 11750

dion2330004@itpln.ac.id

https://doi.org/10.37339/e-komtek.v9i1.2352

Published by Politeknik Piksi Ganesha Indonesia

Artikel Info
Submitted:
03-03-2025
Revised:
05-05-2025
Accepted:
05-06-2025
Online first :
30-06-2025

Abstract

This research presents a single-image dehazing method that integrates a Lightweight Vision Transformer (LVT) and U-Net to capture both local and global features. LVT enhances resolution, U-Net extracts local features, and LVT refines global dependencies before fusion. Evaluations on O-Haze and HSTS datasets show PSNR scores of 27.88 (O-Haze, ResNet-50) and 28.22 (HSTS, no backbone), outperforming existing methods while maintaining competitive SSIM. The results demonstrate effectiveness in real-world haze scenarios, such as wildfire-induced haze in Indonesia.

Keywords: ***Single Image Dehazing; Lightweight Vision Transformer*; U-Net**

Abstrak

Penelitian ini menghadirkan metode dehazing citra tunggal yang mengintegrasikan Lightweight Vision Transformer (LVT) dan U-Net untuk menangkap fitur lokal dan global. LVT meningkatkan resolusi, U-Net mengekstraksi fitur lokal, dan LVT menyempurnakan dependensi global sebelum proses fusi. Evaluasi pada dataset O-Haze dan HSTS menunjukkan skor PSNR sebesar 27.88 (O-Haze, ResNet-50) dan 28.22 (HSTS, tanpa backbone), melampaui metode yang ada sambil mempertahankan skor SSIM yang kompetitif. Hasil ini menunjukkan efektivitas metode dalam skenario kabut dunia nyata, seperti kabut akibat kebakaran hutan di Indonesia.

Kata-kata kunci*: *Single Image Dehazing; Lightweight Vision Transformer; U-Net

1. Introduction

Thick fog and smoke from forest fires, as well as air pollution, can significantly disrupt imaging systems by reducing image clarity **[1].** If left unaddressed, this issue can have serious consequences, particularly in applications that rely on visual analysis, such as autonomous vehicle navigation, traffic monitoring, and surveillance systems. Fog obscures critical details, leading to object misidentification, increased accident risks, and erroneous decisions in automated systems. Technically, airborne particles scatter light, reducing contrast and distorting images before they reach the camera sensor **[2].** For instance, Indonesia's 2023 wildfires affected 994,313.18 hectares across 11 provinces, producing thick smoke that severely degraded image quality and hindered visual analysis processes **[3].** Therefore, effective solutions are required to counteract image degradation caused by fog and smoke.

Dehazing techniques aim to remove haze from images to enhance visibility. Traditional methods, such as image enhancement and physical modeling, still face limitations under varying atmospheric conditions **[4], [5], [6].** Deep learning-based dehazing approaches, particularly those using Convolutional Neural Networks (CNNs) **[7]** and Generative Adversarial Networks (GANs) [8], have improved dehazing quality but struggle to capture global contextual information effectively. Recently, Vision Transformers (ViTs) have been introduced due to their superior ability to model global dependencies. Models like DehazeFormer have proven more efficient than FFA-Net, while DeHamer integrates CNN and Transformer architectures to enhance accuracy **[9], [10].** Additionally, depth-awareness-based approaches, such as DDRL and DehazeDP, leverage depth information to improve results **[11], [12], [13],** whereas depth-agnostic methods are designed for greater flexibility across different conditions [14]. Hybrid models combining traditional image processing with deep learning techniques have further enhanced image clarity **[15], [16],** while physics-based approaches reinforced by neural networks offer promising alternatives **[17].** These advancements not only improve dehazing quality but also pave the way for more efficient real-time applications through optimized end-to-end models and reinforcement learning-driven adaptive strategies **[10], [12].** Comprehensive surveys continue to provide insights into the evolution of dehazing methods, highlighting the strengths and limitations of various approaches **[18], [19], [20], [21], [22].**

The proposed method integrates Lightweight Vision Transformer (LVT) and U-Net for single-image dehazing. This approach employs LVT-based super-resolution before feature extraction. U-Net effectively captures local features, while LVT extracts global contextual information. Experiments explore the impact of different architectural choices, including the use of a pre-trained ResNet-50 backbone versus a model without a backbone. This method aims to balance computational efficiency and image quality, providing a promising direction for future dehazing applications.

2. Method

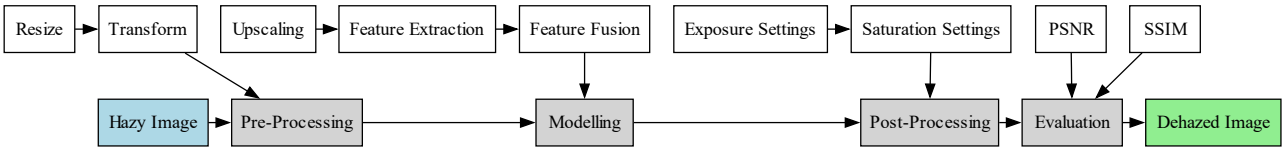

Figure 1. Flowchart of Proposed Dehazing Tehcnique

This research follows the waterfall methodology, consisting of five sequential steps. The process begins with pre-processing, where hazy images (550x441 pixels) are resized and converted into tensors. Next, in the modeling phase, the images undergo upscaling, followed by feature extraction, which is performed both with and without a pre-trained ResNet-50 backbone, and concludes with feature fusion. After modeling, the post-processing stage applies adjustments to exposure and saturation settings to enhance the dehazed images. Each step follows a structured sequence, ensuring a logical transition between phases while maintaining computational efficiency. For a clearer understanding of the research workflow, **Figure 1** provides a flowchart illustrating the process.

2.1 Dataset

2.1.1 Training dataset

The training dataset consists of 1,988 images, including both hazy images and their corresponding ground truth (haze-free) images. This dataset is a secondary dataset sourced from the publicly available SOTS (Synthetic Outdoor Testing Set) **[23],** which is used to train the model to handle real-world conditions where images are affected by haze. All images in this dataset have a resolution of 550x441 pixels. A sample selection from the SOTS dataset is **Figure 2**.

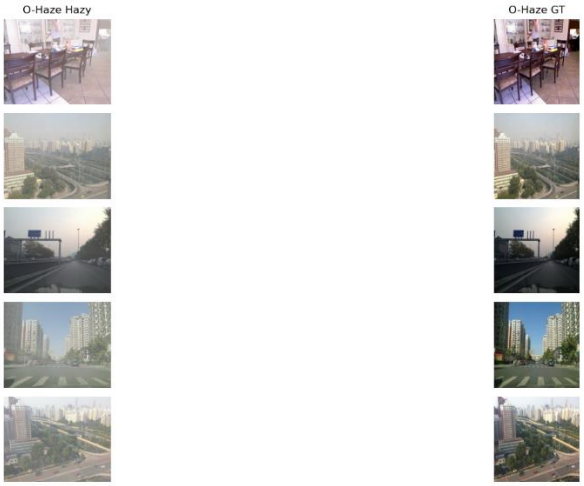

Figure 2. Sampel of SOTS dataset

2.1.2 Testing dataset

Figure 3. Sampel of O-Haze and HSTS dataset

For model testing, two datasets are used: O-Haze, which consists of 45 images **[24],** and HSTS, which contains 10 images **[23].** Both datasets are secondary datasets sourced from publicly available datasets and are used to evaluate the model's performance under varied haze conditions. This assessment helps determine the model's ability to produce clearer images after the dehazing process. All images in these datasets have a resolution of 550x441 pixels. The sample of O-Haze and HSTS dataset can be seen on **Figure 3.**

2.2 Pre-processing

Pre-processing is a crucial step in preparing input data for this research, ensuring that the model receives data in the appropriate format and size for optimal performance **[25].** This stage consists of the following steps:

2.2.1 Resizing

The input images are resized from 550×441 pixels to 256×256 pixels. This step is essential for several reasons:

1. Fixed Input Size Requirement – The model requires a consistent input size to ensure uniform feature extraction across all samples.
2. Computational Efficiency – Reducing the image dimensions lowers memory usage and speeds up both training and inference.
3. Consistent Feature Extraction – Standardizing the input size ensures that all images contribute equally to feature representation learning, preventing bias caused by varying

aspect ratios.

2.2.2 Transform to Tensor

After resizing, the images are converted into tensors. This transformation is necessary for the following reasons:

Framework Compatibility – The deep learning framework used in this research processes data in tensor format rather than traditional image files.

Normalization and Batch Processing – Converting images to tensors enables efficient normalization and batch processing, ensuring a more stable training and inference process.

Channel Arrangement – Tensors structure image data in the format (C, H, W), where C represents the number of channels, and H and W denote the image height and width, respectively.

2.3 Modeling

2.3.1 Model Architecture

2.3.1.1 Upscaling

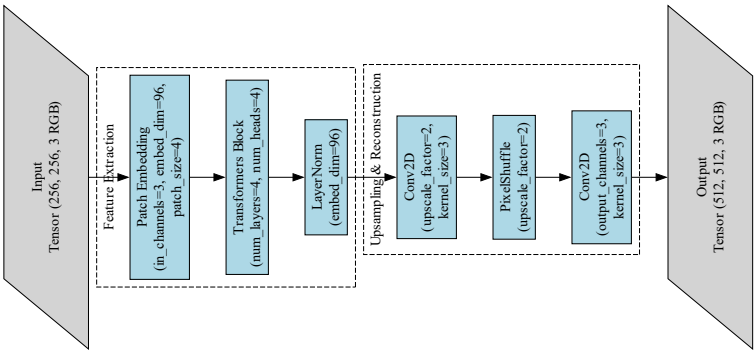

Figure 4. Super-resolution architecture using LVT (Lightweight Vision Transformer)

Upscaling serves as the initial stage in model development, where super-resolution techniques are applied to enhance image quality before the dehazing process takes place. Increasing resolution allows the model to preserve more spatial information, reduce artifacts caused by haze, and facilitate feature extraction in subsequent stages. As shown in Figure 4, the upscaling process begins with a low-resolution input image, which is then enhanced using super-resolution methods to increase its clarity and detail. This improvement in image quality makes the dehazing process more effective, as higher resolution images allow for better estimation of transmission and visibility restoration. Previous research has demonstrated that super-resolution can improve transmission estimation in dehazing, enhance visibility, and produce clearer images with less noise **[26].** By referring to Figure 4, you can observe how the upscaling process enhances

the image before further processing steps are applied. This approach forms the foundation for constructing a more optimal image processing pipeline.

A Lightweight Vision Transformer (LVT) is employed in the super-resolution process to balance feature extraction effectiveness and computational efficiency. Compared to CNN-based methods, LVT leverages self-attention to better capture spatial relationships within the image, enabling more accurate resolution enhancement. The upscaling process begins with a patch embedding layer, which divides the input image into small patches and maps them into a higher-dimensional space to capture fine details. The Transformer block then applies multi-head self-attention (MHSA), allowing the model to focus on various regions within the image and extract both local and global features essential for reconstruction. The upsampling module reconstructs the image at a higher resolution through multiple stages: an initial convolution layer increases the number of feature map channels to match the upscale factor (×2 in this case), followed by PixelShuffle, which rearranges the channels into a higher-resolution format. A final convolution layer refines the reconstructed image quality, ensuring sharper details and improved clarity.

By integrating super-resolution techniques into the upscaling stage, the model can generate high-quality images before undergoing dehazing. This approach preserves crucial details, enabling the model to remove haze more effectively without compromising computational efficiency.

2.3.1.2 Feature extraction

Local feature extraction (U-Net)

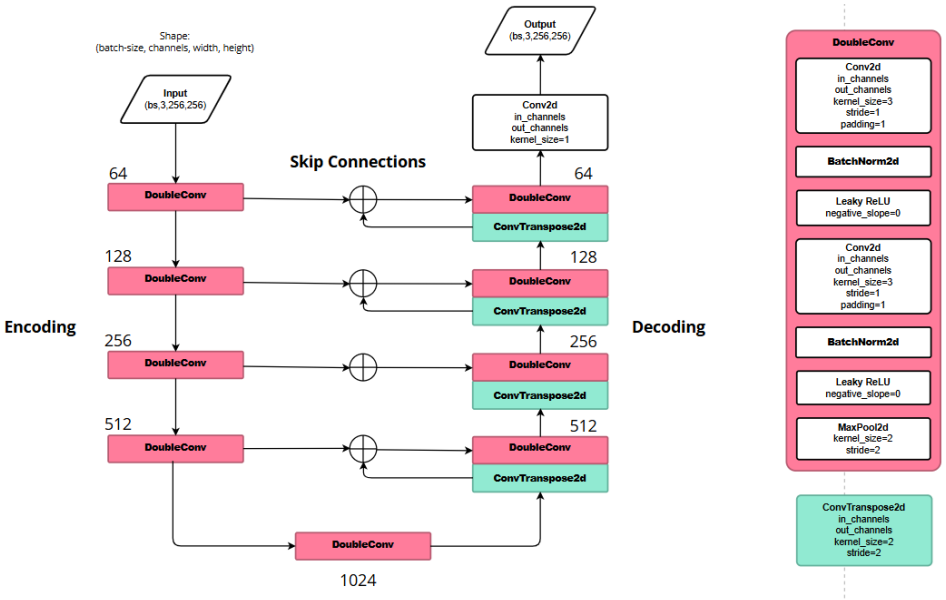

Figure 5. The U-Net architecture used

U-Net was chosen as the primary architecture for dehazing due to its ability to capture fine-grained local features while preserving the spatial structure of the image. This architecture is designed with skip connections that allow information from the initial layers to be retained throughout the later stages, minimizing detail loss during processing. As shown in Figure 5, the U-Net architecture is illustrated, demonstrating its encoder-decoder structure with skip connections that facilitate the retention of essential image details. U-Net provides a balance between accuracy and computational efficiency, making it an optimal choice for this task. By referring to Figure 5, you can observe how the architecture processes and reconstructs the image, ensuring effective dehazing while maintaining critical spatial and local features.

Feature extraction in U-Net is performed through a series of DoubleConv blocks. These blocks are specifically designed to capture fine details from the input image by progressively learning more complex features at different resolution levels. A DoubleConv block consists of two consecutive convolutional operations, each followed by a ReLU activation function. This setup enables the network to extract both low-level and high-level features while maintaining spatial resolution. The DoubleConv block starts with a 3×3 convolution layer followed by ReLU activation, then another 3×3 convolution and ReLU activation. Padding (padding=1) is applied to the convolutions to ensure that the spatial dimensions of the input remain unchanged, which is crucial for preserving feature map resolution. The sequential application of convolutions helps the network capture increasingly complex features.

The dehazing technique in this study employs a U-Net architecture that processes images with channel dimensions ranging from 64 to 1024, allowing the network to capture haze-related information at various levels of abstraction. In U-Net, the network starts with a lower number of channels in the initial layers (64 channels), and the number of channels increases as the network deepens, reaching up to 1024 channels in the deepest layer. Specifically, the first DoubleConv block receives an input image with 64 channels, and subsequent blocks progressively increase the number of channels to capture more complex features. For instance, the second DoubleConv block outputs 128 channels, the third produces 256, and so on, until the deepest block, which generates 1024 channels **[27].** This increasing network depth (from 64 to 1024 channels) enables the model to learn higher-level and more abstract features as the spatial dimensions of feature maps shrink due to pooling operations, which help capture broader contextual information.

Global feature extraction (LVT)

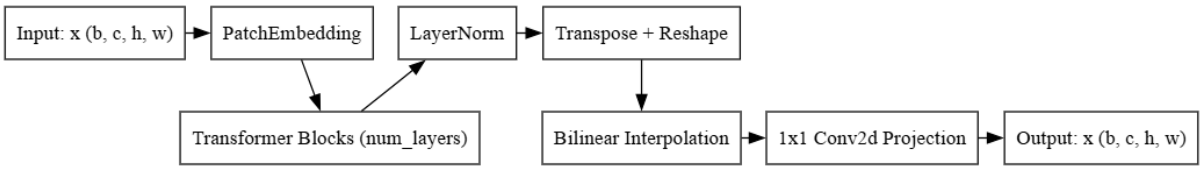

Figure 6. Global Feature Extraction Architecture Using LVT

The global feature extraction model is designed to extract global features from input images by leveraging a Lightweight Vision Transformer (LVT)-based architecture. This architecture, illustrated in **Figure 6,** consists of several key components, including patch embedding, Transformer blocks, layer normalization, and a projection layer, which work together to capture better spatial and semantic representations.

The patch embedding module converts an image into a patch representation before further processing by the Transformer. The in_channels parameter (default: 3) represents the number of input channels, typically 3 for RGB images. The embed_dim parameter (default: 96) determines the dimensionality of the patch representation after embedding, where a higher value allows the model to capture more details but increases computational complexity. Meanwhile, the patch_size parameter (default: 4) defines the size of the patches extracted from the input image, affecting the amount of spatial information retained in each patch.

The global feature extraction process begins with the patch embedding layer, which converts the input image into a feature representation in the form of tokens. This representation is then processed through Transformer blocks, each consisting of multiple self-attention layers to capture long-range dependencies within the image. The model employs num_layers (default: 4), which determines the number of Transformer blocks applied sequentially, with each block containing num_heads (default: 4) for the multi-head self-attention (MHSA) mechanism.

After passing through the Transformer blocks, the extracted features undergo layer normalization to enhance training stability and convergence. The output is then transformed back into a spatial form using transpose and view operations before its resolution is restored to the original size using bilinear interpolation (nn.functional.interpolate(x, size=(h, w), mode='bilinear', align_corners=True)). The final step is the projection layer, which uses nn.Conv2d(embed_dim, in_channels, kernel_size=1) to convert the extracted features back to the same number of channels as the input.

2.4 Post processing

The image tensor is adjusted to the range [0, 1] and then converted to the appropriate byte format for the OpenCV library. The image undergoes exposure enhancement to adjust contrast and brightness, followed by a saturation effect that boosts the red channel and reduces the blue channel. Finally, the image is converted back to RGB format using the PIL library.

2.5 Evaluation

The evaluation was conducted to measure the model's performance using two key metrics: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). The testing was performed on two datasets: O-Haze, which consists of 45 images, and HSTS, which contains 10 images

2.5.1 PSNR

PSNR is a metric used to assess image quality by comparing the processed image with a reference image that is free from distortions. A higher PSNR value indicates better image quality. It is calculated using the following formula:

$$PSNR = 10 \times log_{10} (R^2 / MSE)$$

$$MSE = (1 / MN) \sum \sum [I(i,j) - K(i,j)]^2 [1]$$

where:

- R is the maximum pixel value (e.g., 255 for an 8-bit image)..
- MSE represents the average error between the processed image and the reference image..
- M and N are the dimensions of the image.
- $I(i,j)$ and $K(i,j)$ epresent the pixel intensities at coordinate (i,j)(i,j)(i,j) in the processed and reference images, respectively.

This formula measures the difference between the processed image and the reference image, where a higher PSNR value indicates better image quality.

2.5.2 SSIM

SSIM is a metric that evaluates image quality based on human perception of structure, luminance, and contrast. The SSIM value ranges from -1 to 1, where 1 indicates perfect similarity between the processed image and the reference image. SSIM is calculated using the following formula:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad [30]$$

This formula measures the structural similarity between the processed image and the reference image, where a higher SSIM value indicates better image quality.

2.6 `Experiments

The evaluation aims to assess the effectiveness of feature extraction using a pre-trained backbone (ResNet-50) compared to training from scratch. This approach examines the extent to which previously learned features enhance dehazing performance compared to features extracted without pre-trained weights. Testing is conducted in two main scenarios: (1) without a pre-trained backbone, where feature extraction is performed entirely from scratch, and (2) with a pre-trained backbone (ResNet-50), where features are extracted using ResNet-50 trained on ImageNet before being passed to the U-Net model. In both approaches, the extracted features serve as input to U-Net for generating dehazed images. A comparison is made to evaluate the impact of pre-trained feature extraction on reconstruction quality, training efficiency, and overall model performance.

2.7 Training strategy

The model was trained for 1000 epochs with a batch size of 16 to ensure optimal convergence, using a learning rate of 0.001. The training process employed a combined loss function consisting of MSE Loss, MS-SSIM Loss, and Perceptual Loss, each weighted accordingly. This approach was designed to balance direct pixel-wise errors, preserve spatial structure, and enhance the perceptual quality of the dehazed images.

The loss function used is defined as follows:

$$L\text{total} = \alpha \cdot L\text{MSE} + (1 - \alpha) \cdot L\text{MS-SSIM} + \beta \cdot L\text{Perceptual}$$

With $LMSE$ as the Mean Squared Error (MSE) loss, which measures the difference between the output and the reference image, the hyperparameters are set as follows: α=0.84 to emphasize the MSE loss, while Perceptual Loss is weighted at 0.15 to preserve the perceptual quality of the dehazed image. MS-SSIM loss is assigned a weight of α, which is 0.16, to ensure that the spatial structure remains intact. By combining these loss functions, the model is able to produce images with not only low pixel error but also improved detail preservation and perceptual quality.

3. Results and Discussion

3.1 Qualitative Result

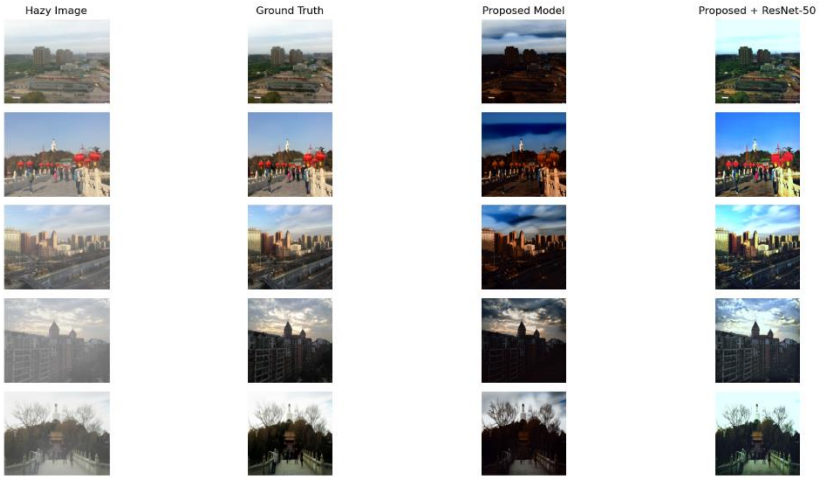

Figure 7. The Sample of Dehazing Result for HSTS Dataset

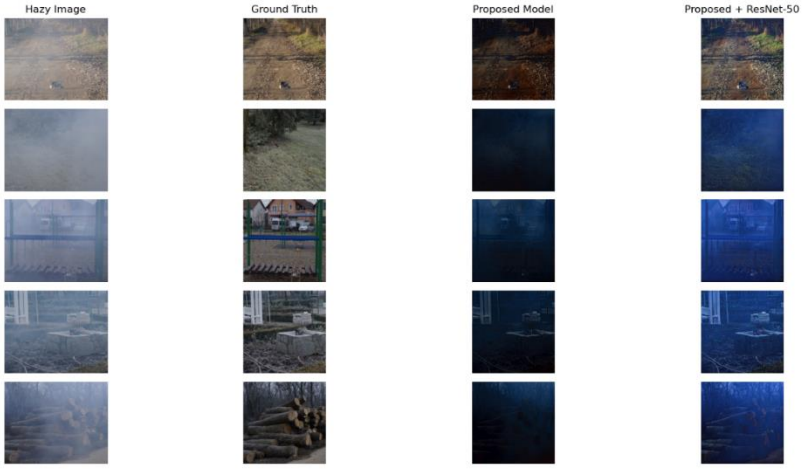

Figure 8. The sample of Dehazing Result for O-Haze Dataset

The qualitative results of the proposed method reveal challenges such as unnatural color tints, notably a darker blue hue, especially when the model does not fully restore colors. Residual haze in certain areas suggests incomplete haze removal, potentially due to limited diversity in the training data, which may not encompass all haze conditions. The dehazing results for the HSTS dataset are shown in Figure 7, and those for the O-Haze dataset are presented in Figure 8.

Adjustments to the architecture or loss function, along with incorporating a broader range of haze conditions in the training data, have been shown in other studies to improve both color accuracy and consistency in haze removal. For instance, Li et al. [31] proposed a nighttime image dehazing method that addresses severe color distortion and complex lighting conditions by integrating color cast removal with a dual-path multi-scale fusion algorithm. Similarly, Dutta et

al. [32] enhanced degraded nighttime images through a combination of dehazing and color correction techniques, effectively improving color consistency.

3.2 Quantitative Result

Tabel 1. Average of PSNR and SSIM for HSTS and O-Haze dataset

Dataset	Metric	Proposed	Proposed + ResNet-50	Kim, 2021	Salazar-Colores, et al., 2022	Hartanto dan Rahdianti, 2021
O-Haze	PSNR	27.8	27.88	15.78	-	25.39
	SSIM	0/61	0.64	0.49	-	0.79
HSTS	PSNR	28.22	28.09	22.36	24.49	-
	SSIM	0.77	0.81	0.9	0.9	-

The presented results that shown on Table 1. evaluate various image dehazing methods using PSNR and SSIM metrics on two datasets: O-Haze and HSTS. The table compares the performance of the proposed model with other methods, including Kim (2021) **[33],** Salazar-Colores et al. (2022) **[34],** Hartanto & Rahdianti (2021) **[35],** and Roy & Chaudhuri (2024) **[26].**

***Peak signal to noise ratio* (PSNR)**

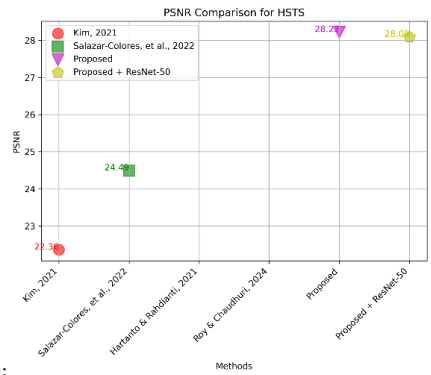

Figure 3. Comparison of PSNR value for HSTS dataset

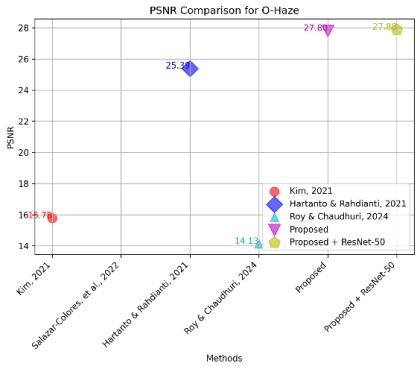

Figure 4. Comparsion of SSIM value for O-Haze dataset

The PSNR values for the dehazing results are shown in Figure 9 for the HSTS dataset and Figure 10 for the O-Haze dataset. For the O-Haze dataset, the proposed method with a ResNet-50 backbone achieved the highest PSNR of 27.88, slightly outperforming the model without a backbone (27.8). Other methods performed worse, like Kim (2021) with 15.78, Roy & Chaudhuri (2024) with 14.13, and Hartanto & Rahdianti (2021) with 25.39.

For the HSTS dataset, the best performance came from the proposed model without a backbone, achieving a PSNR of 28.22, surpassing other methods like Kim (2021) (22.36) and

Salazar-Colores et al. (2022) (24.49). A higher PSNR indicates better dehazing quality, with lower noise and distortion, bringing the output image closer to the ground truth [29].

***Structural Similarity Index Measure* (SSIM)**

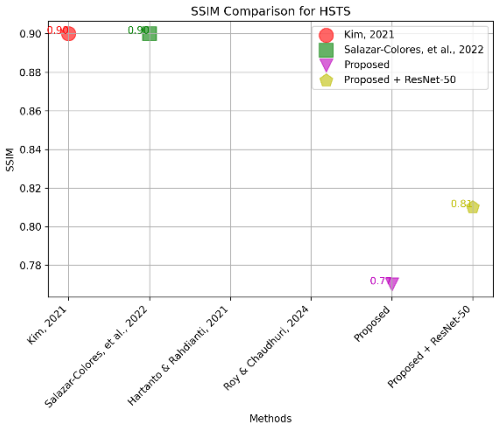

Figure 6. Comparison of SSIM value for HSTS dataset

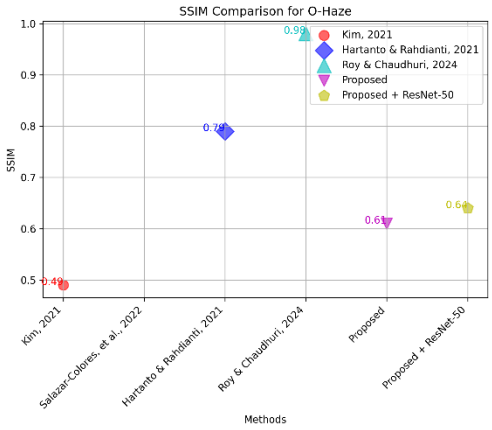

Figure 5. Comparison of SSIM value for O-Haze dataset

The SSIM values for the dehazing results are shown in Figure 11 for the HSTS dataset and Figure 12 for the O-Haze dataset. For the O-Haze dataset, the proposed model without a backbone achieved an SSIM of 0.61, which improved to 0.64 with the ResNet-50 backbone. Other methods showed varying results, such as Kim (2021) with 0.49, Hartanto & Rahdianti (2021) with 0.79, and Roy & Chaudhuri (2024) with 0.98.

For the HSTS dataset, the proposed model achieved an SSIM of 0.77 without a backbone and 0.81 with ResNet-50. However, Kim (2021) and Salazar-Colores et al. (2022) achieved higher SSIM values of 0.9. While the proposed model shows improvements in PSNR, the SSIM results suggest that the spatial structure of the dehazed images can still be further enhanced.

Overall, the proposed approach demonstrates superior performance in improving image quality based on PSNR, effectively reducing noise and distortion compared to other methods. However, since the SSIM metric remains lower than some competing methods, there is still room for improvement in preserving the spatial structures of dehazed images.

4. Conclusion

This research proposes a single-image dehazing method that integrates Lightweight Vision Transformer (LVT) and U-Net to enhance hazy image quality. The approach utilizes LVT for super-resolution, U-Net for local feature extraction, and LVT again for global feature extraction before merging the results. Experimental evaluations on the O-Haze and HSTS datasets

demonstrate the method's effectiveness, achieving a PSNR of 27.88 with a ResNet-50 backbone and 28.09 without a backbone. The proposed approach successfully mitigates image degradation caused by haze, including smoke haze from forest fires in Indonesia. For further improvements, incorporating a more diverse dataset can enhance generalization across different haze conditions, while optimizing the model architecture and exploring more suitable loss functions may further refine performance. Additionally, real-time implementation should be considered to enable applications in surveillance systems, remote sensing, and autonomous navigation in foggy environments.

References

[1] X. Jin, R. Tang, L. Liu, and J. Wu, "Vehicle license plate recognition for fog-haze environments," *IET Image Process*, vol. 15, no. 6, pp. 1273–1284, 2021, doi: https://doi.org/10.1049/ipr2.12103.

[2] F. Guo, J. Yang, Z. Liu, and J. Tang, "Haze removal for single image: A comprehensive review," *Neurocomputing*, vol. 537, pp. 85–109, 2023, doi: https://doi.org/10.1016/j.neucom.2023.03.061.

[3] Kementrian Lingkungan Hidup dan Kehutanan, "Kinerja pengendalian kebakaran hutan dan lahan tahun 2023," *https://ppid.menlhk.go.id/berita/siaran-pers/7579/kinerja-pengendalian-kebakaran-hutan-dan-lahan-tahun-2023*, 2024.

[4] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1956–1963. doi: 10.1109/CVPR.2009.5206515.

[5] O. D. Nurhayati, B. Surarso, W. A. Syafei, and D. M. K. Nugraheni, "Gaussian filter-based dark channel prior for image dehazing enhancement," *International Journal of Electrical and Computer Engineering*, vol. 14, no. 5, pp. 5765–5778, Oct. 2024, doi: 10.11591/ijece.v14i5.pp5765-5778.

[6] W. Yan and L. Cui, "Image Dehaze Algorithm Based on Improved Atmospheric Scattering Models," *IEEE Access*, vol. 12, pp. 98971–98976, 2024, doi: 10.1109/ACCESS.2024.3428568.

[7] E. Wang, S. Shu, and C. Fan, "CNN-based Single Image Dehazing via Attention Module," in *2022 IEEE 5th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, 2022, pp. 683–687. doi: 10.1109/AUTEEE56487.2022.9994347.

[8] A. Zhao, L. Li, and S. Liu, "UIDF-Net: Unsupervised Image Dehazing and Fusion Utilizing GAN and Encoder–Decoder," *J Imaging*, vol. 10, no. 7, 2024, doi: 10.3390/jimaging10070164.

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

[10] C. Guo, Q. Yan, S. Anwar, R. Cong, W. Ren, and C. Li, "Image Dehazing Transformer with Transmission-Aware 3D Position Embedding," in *2022 IEEE/CVF Conference on*

Computer Vision and Pattern Recognition (CVPR), 2022, pp. 5802–5810. doi: 10.1109/CVPR52688.2022.00572.

[11] T. Guo and V. Monga, "Reinforced Depth-Aware Deep Learning for Single Image Dehazing," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8891–8895. doi: 10.1109/ICASSP40776.2020.9054504.

[12] Z. Liu, B. Xiao, M. Alrabeiah, K. Wang, and J. Chen, "Single Image Dehazing with a Generic Model-Agnostic Convolutional Neural Network," *IEEE Signal Process Lett*, vol. 26, no. 6, pp. 833–837, 2019, doi: 10.1109/LSP.2019.2910403.

[13] M. A.-N. I. Fahim and H. Y. Jung, "Single Image Dehazing Using End-to-End Deep-Dehaze Network," *Electronics (Basel)*, vol. 10, no. 7, 2021, doi: 10.3390/electronics10070817.

[14] Y. Zhang and Y. Dong, "Single Image Dehazing via Reinforcement Learning," in *2020 IEEE International Conference on Information Technology,Big Data and Artificial Intelligence (ICIBA)*, 2020, pp. 123–126. doi: 10.1109/ICIBA50161.2020.9277382.

[15] G. Kim, J. Park, and J. Kwon, "Deep Dehazing Powered by Image Processing Network," 2023.

[16] N. Jiang, K. Hu, T. Zhang, W. Chen, Y. Xu, and T. Zhao, "Deep hybrid model for single image dehazing and detail refinement," *Pattern Recognit*, vol. 136, p. 109227, 2023, doi: https://doi.org/10.1016/j.patcog.2022.109227.

[17] Z. Li, C. Zheng, H. Shu, and S. Wu, "Single Image Dehazing via Model-Based Deep-Learning," in *2022 IEEE International Conference on Image Processing (ICIP)*, 2022, pp. 141–145. doi: 10.1109/ICIP46576.2022.9897479.

[18] J. Gui *et al.*, "A Comprehensive Survey and Taxonomy on Single Image Dehazing Based on Deep Learning," *ACM Comput. Surv.*, vol. 55, no. 13s, Jul. 2023, doi: 10.1145/3576918.

[19] Y. Kang, L. Zhang, P. Hu, Y. Liu, H. Lu, and Y. He, "Learning depth-aware decomposition for single image dehazing," *Computer Vision and Image Understanding*, vol. 248, p. 104069, 2024, doi: https://doi.org/10.1016/j.cviu.2024.104069.

[20] S. R. Gumma and B. M. Chintakindi, "FoNet: Focused Network for Single Image Deraining," *Circuits Syst Signal Process*, 2025, doi: 10.1007/s00034-025-03009-9.

[21] D. Rawat and K. Singh, "A Comparative Study on Single Image Dehazing using Deep Learning-Based Techniques," in *2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, 2023, pp. 783–789. doi: 10.1109/ICAC3N60023.2023.10541689.

[22] L. Li and Y. Ning, "A review of image dehazing based on deep learning," 2024. [Online]. Available: www.ijerm.com

[23] B. Li *et al.*, "Benchmarking Single-Image Dehazing and Beyond," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 492–505, 2019.

[24] C. O. Ancuti, C. Ancuti, R. Timofte, and C. De Vleeschouwer, "O-HAZE: a dehazing benchmark with real hazy and haze-free outdoor images," in *IEEE Conference on Computer Vision and Pattern Recognition, NTIRE Workshop* , in NTIRE CVPR'18. 2018.

[25] D. Murcia-Gómez, I. Rojas-Valenzuela, and O. Valenzuela, "Impact of Image Preprocessing Methods and Deep Learning Models for Classifying Histopathological Breast Cancer Images," *Applied Sciences*, vol. 12, no. 22, 2022, doi: 10.3390/app122211375.

[26] S. Roy and S. Chaudhuri, "SIVDSR-Dhaze: Single Image Dehazing with Very Deep Super Resolution Framework and Its Analysis," *Scientific Visualization*, vol. 14, Feb. 2022, doi: 10.26583/sv.14.5.02.

[27] H. Zhou, Z. Chen, Q. Li, and T. Tao, "Dehaze-UNet: A Lightweight Network Based on UNet for Single-Image Dehazing," *Electronics (Basel)*, vol. 13, no. 11, 2024, doi: 10.3390/electronics13112082.

[28] N. Gawande, D. Goyal, and K. Sankhla, "Improved Deep Learning and Feature Fusion Techniques for Chronic Heart Failure," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 17s, pp. 67–80, 2024, [Online]. Available: https://www.ijisae.org/index.php/IJISAE/article/view/4837

[29] R. Dhivya and N. Shanmugapriya, "An Analysis Study of Various Image Preprocessing Filtering Techniques based on PSNR for Leaf Images," in *2022 International Conference on Advanced Computing Technologies and Applications (ICACTA)*, 2022, pp. 1–8. doi: 10.1109/ICACTA54488.2022.9753444.

[30] S. Lee, S. Hong, G. Kim, and J. Ha, "SSIM-Based Autoencoder Modeling to Defeat Adversarial Patch Attacks," *Sensors*, vol. 24, no. 19, 2024, doi: 10.3390/s24196461.

[31] B. Wang, L. Hu, B. Wei, Z. Kang, and C. Li, "Nighttime image dehazing using color cast removal and dual path multi-scale fusion strategy," *Front Comput Sci*, vol. 16, no. 4, p. 164706, 2021, doi: 10.1007/s11704-021-0162-x.

[32] R. Lenka, A. Khandual, K. Dutta, and S. Nayak, "Image Enhancement: Application of Dehazing and Color Correction for Enhancement of Nighttime Low Illumination Image," 2019, pp. 211–223. doi: 10.4018/978-1-7998-0066-8.ch011.

[33] C. Kim, "Region Adaptive Single Image Dehazing," *Entropy*, vol. 23, no. 11, 2021, doi: 10.3390/e23111438.

[34] S. Salazar-Colores, E. U. Moya-Sánchez, J.-M. Ramos-Arreguín, E. Cabal-Yépez, G. Flores, and U. Cortés, "Fast Single Image Defogging With Robust Sky Detection," *IEEE Access*, vol. 8, pp. 149176–149189, 2020, doi: 10.1109/ACCESS.2020.3015724.

[35] C. A. Hartanto and L. Rahadianti, "Single Image Dehazing Using Deep Learning," *JOIV : International Journal on Informatics Visualization*, 2021, doi: http://dx.doi.org/10.30630/joiv.5.1.431.