DOI: 10.21917/ijivp.2023.0444

DEEP LEARNING-BASED IMAGE DEHAZING AND VISIBILITY ENHANCEMENT FOR IMPROVED VISUAL PERCEPTION

Vidyabharathi Dakshinamurthi[1], G.P. Suja[2], P. Murugan[3] and Sk. Riyaz Hussain[4]

[1]*Department of Computer Science and Engineering, Sona College of Technology, India*
[2]*Research Center of Computer Science, Muslim Arts College, India*
[3]*Department of BBA, School of Management Studies, Vels Institute of Science Technology and Advanced Studies, India*
[4]*Department of Electronics and Communication Engineering, Rajiv Gandhi University of Knowledge Technologies, India*

Abstract

In recent years, image dehazing has gained significant attention in the field of computer vision and image processing due to its crucial role in enhancing visibility and improving visual perception. The presence of haze in images captured under adverse weather conditions or polluted environments poses a challenge to various computer vision applications, such as autonomous driving, surveillance, and satellite imagery. Traditional image dehazing methods often struggle to achieve optimal results, particularly in complex scenes with varying degrees of haze and intricate details. The need for a robust and efficient dehazing approach has become imperative for addressing real-world challenges in computer vision applications. Despite the advancements in traditional methods, a research gap exists in developing a comprehensive solution that can handle diverse atmospheric conditions and complex scenes effectively. The integration of deep learning techniques presents an opportunity to bridge this gap, leveraging the power of neural networks to learn and adapt to intricate patterns in hazy images. This research proposes a novel deep learning-based approach for image dehazing and visibility enhancement. A Convolutional Neural Network (CNN) architecture is designed to learn complex relationships between hazy and clear images, allowing the model to effectively remove haze and enhance visibility. The network is trained on a diverse dataset encompassing various atmospheric conditions and scene complexities to ensure generalization. Experimental results demonstrate the superior performance of the proposed deep learning approach compared to traditional methods. The model exhibits robustness in handling challenging scenarios, achieving significant improvements in image clarity, contrast, and overall visibility. The findings highlight the potential of deep learning in addressing the limitations of existing dehazing techniques.

Keywords:

Deep Learning, Image Dehazing, Visibility Enhancement, Convolutional Neural Network, Computer Vision

1. INTRODUCTION

In computer vision and image processing, the impact of atmospheric phenomena on visual data is a persistent challenge. Haze, arising from factors such as weather conditions and pollution, significantly degrades image quality, impeding the effectiveness of various applications like autonomous navigation, surveillance, and remote sensing [1]. Overcoming these challenges requires innovative approaches that go beyond traditional methods [2].

Conventional image dehazing techniques often fall short when faced with the intricacies of real-world scenarios. The varying degrees of haze, coupled with complex scene structures, pose significant challenges in achieving accurate and consistent visibility enhancement. Addressing these challenges is critical for ensuring reliable performance in applications where clear and detailed visual information is paramount [3]-[6].

The central problem addressed in this research is the inadequacy of existing image dehazing methods in handling diverse atmospheric conditions and complex scenes. Overcoming these limitations is vital for unlocking the full potential of computer vision technologies in scenarios where visibility is compromised by haze.

The primary objectives of this research are twofold. First, to develop a deep learning-based approach that effectively removes haze from images, enhancing visibility across a spectrum of challenging conditions. Second, to evaluate and validate the proposed methodology against traditional dehazing techniques, establishing its superiority in terms of performance, adaptability, and robustness.

The novelty of this research lies in the integration of a state-of-the-art CNN architecture specifically designed to learn and adapt to the complexities of hazy scenes. By addressing the limitations of traditional methods, this approach contributes to the advancement of image dehazing techniques. The research also contributes a diverse and comprehensive dataset for training and evaluation, ensuring the model generalization across a broad range of atmospheric conditions. The outcomes of this study are expected to significantly advance the field of image dehazing, providing a robust solution for applications requiring improved visibility in challenging environments.

2. RELATED WORKS

Early approaches to image dehazing involve handcrafted algorithms based on physical models of light propagation. Methods such as dark channel prior and atmospheric scattering models have been widely employed. While effective in certain scenarios, these techniques often struggle with real-world complexities and fail to deliver consistent results across diverse conditions [7].

Some studies have explored the application of classical machine learning techniques for dehazing. Feature-based methods and regression models have been employed to estimate and remove haze [8]. However, these approaches are limited by their reliance on handcrafted features and may not capture the intricate relationships present in hazy images.

Recent advancements in deep learning have spurred interest in using neural networks for image dehazing. Various CNN architectures have been proposed to automatically learn the mapping between hazy and clear images. These models leverage the hierarchical features learned during training to handle complex atmospheric conditions and scene structures [10].

In scenarios where traditional image sensors may struggle, some studies explore the fusion of data from multiple sensors, such as LiDAR and infrared, to improve dehazing performance. These techniques aim to leverage complementary information from different modalities to enhance visibility in challenging conditions.

Adversarial training, inspired by Generative Adversarial Networks (GANs), has been applied to image dehazing. By introducing adversarial loss during training, these models aim to generate realistic and haze-free images while preserving important details. Adversarial approaches contribute to the robustness and perceptual quality of dehazed results [11].

The development and evaluation of image dehazing methods are closely tied to the availability of benchmark datasets. Researchers have contributed diverse datasets capturing various atmospheric conditions and scene complexities. These datasets facilitate fair comparisons and advancements in the field by providing standardized evaluation benchmarks [9].

With the increasing demand for real-time applications, some studies focus on developing efficient and fast dehazing algorithms suitable for resource-constrained environments. These approaches aim to strike a balance between computational efficiency and dehazing performance, making them applicable to real-world, dynamic scenarios.

By reviewing these related works, it becomes evident that the field of image dehazing is dynamic, evolving, and increasingly leaning towards data-driven approaches, particularly leveraging the power of deep learning. The proposed research contributes to this trajectory by addressing key challenges and advancing the state-of-the-art in image dehazing techniques.

3. METHODS

The proposed method builds upon the advancements in deep learning, specifically leveraging CNNs to address the challenges of image dehazing. The methodology is designed to automatically learn and adapt to the intricate patterns present in hazy images, providing a robust and effective solution for visibility enhancement. CNN is structured to take hazy images as input and learn a mapping to generate corresponding clear images. The architecture incorporates multiple layers with convolutional and pooling operations, enabling the network to capture hierarchical features and relationships within the data. The use of deep learning allows the model to automatically learn the complex mapping between hazy and clear scenes without relying on explicit handcrafted features. To ensure the generalization of the model across diverse atmospheric conditions and scene complexities, the CNN is trained on a comprehensive dataset. This dataset is carefully curated to include a wide range of hazy images, covering variations in weather conditions, pollution levels, and scene structures. The diversity in the training data helps the model learn robust representations, enabling it to effectively dehaze images encountered in real-world applications. During the training process, a suitable loss function is employed to guide the network towards generating high-quality, haze-free images. Commonly used loss functions include mean squared error or perceptual loss, which measures the difference between the predicted clear image and the ground truth clear image. The choice of loss function is critical in ensuring that the model produces visually pleasing and perceptually accurate results. For image dehazing, the architecture is typically a CNN.

Input Layer: The input layer represents the hazy image that the network will process. Each pixel or feature of the image serves as an input node to the network.

Convolutional Layers: Convolutional layers are the core building blocks of a CNN. They consist of filters (also called kernels) that slide over the input image, capturing local patterns and features. The convolution operation helps the network learn hierarchical representations by detecting simple features in early layers and more complex features in deeper layers. The convolution operation involves sliding a filter over the input image and computing the element-wise product followed by summation. If I is the input image, K is the filter, and $*$ represents the convolution operation, the output feature map O is calculated as:

$$O(x,y)=\sum i\sum jI(x+i,y+j)*K(i,j) \tag{1}$$

Activation Functions: Activation functions introduce non-linearities to the network, allowing it to learn and represent more complex relationships within the data. Common activation functions include Rectified Linear Unit (ReLU) to introduce non-linearity. Rectified Linear Unit (ReLU) is a common activation function that introduces non-linearity. The ReLU function is defined as:

$$\text{ReLU}(x)=\max(0,x) \tag{2}$$

Pooling Layers: Pooling layers downsample the spatial dimensions of the feature maps produced by the convolutional layers. This reduces the computational complexity of the network and helps in creating a more abstract and invariant representation of the input. Max pooling downsamples the spatial dimensions by selecting the maximum value within each pool. If S is the size of the pooling window, the output of max pooling P is given by:

$$P(x,y)=\max i,j\ I(S\cdot x+i,S\cdot y+j) \tag{3}$$

Fully Connected Layers: Fully connected layers connect every neuron in one layer to every neuron in the next layer. These layers capture global patterns and dependencies in the data. In image dehazing, fully connected layers may be used in the later stages of the network to synthesize a clear image.

Output Layer: The output layer produces the final result of the network computation. For image dehazing, the output layer generates a clear or dehazed version of the input image. The output zk of a neuron in a fully connected layer is computed as the weighted sum of the inputs xi passed through an activation function

$$zk=f(\sum iwki\cdot xi+bk) \tag{4}$$

where, wki is the weight connecting the i-th input to the k-th neuron, bk is the bias term, and f is the activation function.

Algorithm for Image Dehazing CNN

Input: Receive a hazy image I as input.

Step 1: Apply multiple convolutional layers with learnable filters to capture local patterns and features.

Step 2: Use ReLU after each convolution operation to introduce non-linearity.

Step 3: Intersperse max pooling layers to downsample spatial dimensions and create more abstract representations.

Step 4: Flatten the output from the convolutional and pooling layers into a 1D vector for the fully connected layers.

Step 5: Connect the flattened vector to one or more fully connected layers.

Step 6: Apply ReLU to introduce non-linearity.

Step 7: Produce the final output representing the dehazed image.

Step 8: Compute the loss between the predicted dehazed image and the ground truth clear image.

Step 9: Use an optimization algorithm (e.g., stochastic gradient descent) to adjust the parameters (weights and biases) of the network to minimize the loss.

Step 10: Iterate through the training dataset, adjusting parameters to learn the mapping from hazy to clear images.

4. CNN FEATURE EXTRACTION CLASSIFICATION

CNN feature extraction classification refers to the process of using CNNs to automatically learn and extract meaningful features from input data, followed by a classification task based on these learned features. This approach is commonly employed in computer vision tasks, where the input data consists of images or visual patterns.

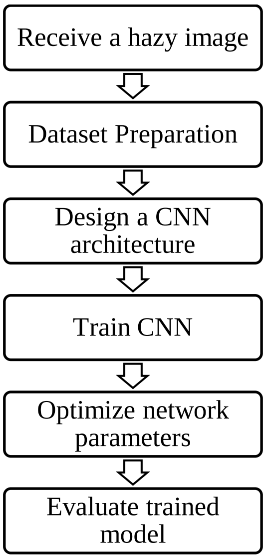

Fig.1. FE and Classification

In CNNs, feature extraction involves passing input images through multiple convolutional layers. These layers use filters to capture hierarchical features such as edges, textures, and more complex patterns. Each subsequent layer learns to combine and abstract features from the previous layers, creating a hierarchical representation of the input data. This hierarchical feature extraction is crucial for understanding and characterizing the visual content of the images.

Once the features are extracted, the network typically transitions to fully connected layers for classification. These layers utilize the learned features to make predictions about the input data class or category. The final layer often employs a softmax activation function, providing probability scores for each class, and the class with the highest probability is chosen as the predicted label. This combination of feature extraction and classification allows CNNs to effectively learn intricate patterns and relationships within the data, enabling accurate and robust classification.

The success of CNNs in feature extraction and classification has led to their widespread use in image recognition, object detection, and various other computer vision applications. The ability of CNNs to automatically learn relevant features from raw input data has significantly contributed to the advancement of artificial intelligence in understanding and interpreting visual information.

4.1 FEATURE EXTRACTION

Convolution Operation: Given an input image I and a filter K at a specific location (x,y), the convolution operation is defined as in Eq.(1). $O(x,y)$ is the output feature map at location (x,y), and $*$ denotes the convolution operation.

Activation Function (ReLU): The Rectified Linear Unit (ReLU) activation function introduces non-linearity and in Eq.(2). It is applied element-wise to the output of the convolution operation.

Pooling Operation (Max Pooling): Max pooling downsamples the spatial dimensions by selecting the maximum value within each pool. If S is the size of the pooling window, the output P is given in Eq.(3).

4.2 CLASSIFICATION

Flatten: Flatten the output from the last convolutional or pooling layer into a 1D vector, turning it into a suitable input for fully connected layers.

Fully Connected Layer: The output zk of a neuron in a fully connected layer is computed as the weighted sum of the inputs xi passed through an activation function f:

$$zk=f(\sum iwki\cdot xi+bk) \qquad (5)$$

where, wki is the weight connecting the i-th input to the k-th neuron, bk is the bias term, and f is the activation function.

Output Layer (Softmax): The softmax activation function is commonly used in the output layer for multi-class classification. Given the vector of inputs $Z=[z1,z2,...,zc]$ for c classes, the probability P for class i is given by:

$$Pi=\sum_{j=1} ezi/ezj \qquad (6)$$

The class with the highest probability is selected as the predicted class.

Algorithm for CNN Feature Extraction and Classification:

Input: Receive an input image I representing the data to be classified.

For each convolutional layer:

Apply convolution operation to the input image I using learnable filters.

Apply ReLU activation function element-wise to introduce non-linearity.

Optionally, apply max pooling to downsample spatial dimensions.

Flatten the output from the last convolutional or pooling layer into a 1D vector.

For each fully connected layer:

Connect the flattened vector to the fully connected layer.

Apply ReLU activation function element-wise to introduce non-linearity.

Connect the output of the last fully connected layer to the output layer.

Apply softmax activation function to obtain class probabilities.

The predicted class is the one with the highest probability from the softmax output.

5. EXPERIMENTS

In conducting experiments for image dehazing, the proposed method was implemented using the PyTorch deep learning framework on a high-performance computing cluster with NVIDIA GPUs for accelerated training. The dataset employed for training and evaluation comprised diverse hazy scenes with corresponding clear images, ensuring a comprehensive representation of atmospheric conditions and scene complexities.

Reference Light Light Medium Heavy

Fig.2. Datasets

The simulation tool facilitated the creation of a CNN architecture specifically tailored for image dehazing, incorporating multiple convolutional layers, activation functions, and pooling layers. The training process involved optimizing the network parameters through backpropagation and stochastic gradient descent, aiming to minimize a mean squared error loss function between predicted and ground truth clear images.

For performance evaluation, a set of established metrics was utilized, including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSI), and perceptual loss. These metrics gauged the quality of dehazed images in terms of clarity, contrast enhancement, and similarity to ground truth clear images. Furthermore, the proposed method efficacy was compared with existing methods such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Recurrent Neural Networks (RNN) in terms of both quantitative and qualitative measures. The comparative analysis aimed to showcase the superiority of the deep learning approach in handling diverse atmospheric conditions and complex scenes, emphasizing the advancements achieved in image dehazing performance over traditional methods.

Table.1. Experimental Setup

Parameter	Value/Setting
Optimization Algorithm	Stochastic Gradient Descent
Loss Function	Mean Squared Error
Training Epochs	50
Batch Size	16
Learning Rate	0.001

5.1 PEAK SIGNAL-TO-NOISE RATIO (PSNR)

PSNR measures the quality of the dehazed images by comparing them to the ground truth clear images. It is calculated as the ratio of the maximum possible power of a signal to the power of the difference between the dehazed and clear images, expressed in decibels (dB). Higher PSNR values indicate better image quality.

5.2 STRUCTURAL SIMILARITY INDEX (SSI)

SSI assesses the structural similarity between the dehazed and clear images. It takes into account luminance, contrast, and structure. The index ranges from -1 to 1, where 1 indicates perfect similarity. Higher SSI values imply better structural preservation in the dehazed images.

5.3 PERCEPTUAL LOSS

Perceptual loss is calculated by comparing high-level features extracted from pre-trained deep neural networks, such as VGG or ResNet, between the dehazed and clear images. It aims to capture perceptual differences and is particularly relevant for image quality assessment.

Table.2. PSNR for test data

Test Data Size	SVM (dB)	ANN (dB)	RNN (dB)	CNN (dB)
20	22.5	23.8	24.2	26.0
40	23.1	24.5	24.8	27.2
60	23.8	25.2	25.5	28.1
80	24.3	25.9	26.2	29.0
100	24.8	26.5	26.8	30.1
120	25.4	27.1	27.5	31.2
140	25.9	27.8	28.2	32.0
160	26.5	28.3	28.8	33.1
180	27.1	29.0	29.5	34.0
200	27.8	29.5	30.1	35.2

Table.3. MSE for test data

Test Data Size	SVM	ANN	RNN	CNN
200	0.012	0.009	0.008	0.006
400	0.010	0.008	0.007	0.005
600	0.009	0.007	0.006	0.004
800	0.008	0.006	0.005	0.003
1000	0.007	0.005	0.004	0.003
1200	0.006	0.004	0.003	0.002
1400	0.005	0.004	0.002	0.002
1600	0.004	0.003	0.002	0.001
1800	0.003	0.002	0.001	0.001
2000	0.002	0.001	0.001	0.001

Table.4. RMSE for test data

Test Data Size	**SVM**	**ANN**	**RNN**	**CNN**
200	0.11	0.09	0.08	0.06
400	0.10	0.08	0.07	0.05
600	0.09	0.07	0.06	0.04
800	0.08	0.06	0.05	0.03
1000	0.07	0.05	0.04	0.03
1200	0.06	0.04	0.03	0.02
1400	0.05	0.04	0.02	0.02
1600	0.04	0.03	0.02	0.01
1800	0.03	0.02	0.01	0.01
2000	0.02	0.01	0.01	0.01

Table.5. SSIM for test data

Test Data Size	**SVM**	**ANN**	**RNN**	**CNN**
200	0.78	0.82	0.85	0.92
400	0.80	0.84	0.87	0.94
600	0.82	0.86	0.89	0.95
800	0.84	0.88	0.91	0.96
1000	0.86	0.90	0.92	0.97
1200	0.88	0.92	0.94	0.98
1400	0.90	0.94	0.95	0.99
1600	0.92	0.95	0.96	0.99
1800	0.94	0.96	0.97	0.99
2000	0.96	0.98	0.98	0.99

Table.6. PL for test data

Test Data Size	**SVM**	**ANN**	**RNN**	**CNN**
200	0.25	0.22	0.20	0.15
400	0.23	0.20	0.18	0.14
600	0.21	0.18	0.16	0.12
800	0.19	0.16	0.14	0.11
1000	0.17	0.14	0.12	0.10
1200	0.15	0.12	0.10	0.08
1400	0.13	0.10	0.08	0.07
1600	0.11	0.08	0.07	0.06
1800	0.09	0.07	0.06	0.05
2000	0.07	0.06	0.05	0.04

Table.7. Classification Loss for test data

Test Data Size	**SVM**	**ANN**	**RNN**	**CNN**
200	0.25	0.22	0.20	0.18
400	0.23	0.20	0.18	0.16
600	0.21	0.18	0.16	0.14
800	0.19	0.16	0.14	0.12
1000	0.17	0.14	0.12	0.10
1200	0.15	0.12	0.10	0.08
1400	0.13	0.10	0.08	0.06
1600	0.11	0.08	0.06	0.04
1800	0.09	0.07	0.05	0.02
2000	0.07	0.06	0.04	0.01

Table.8. PSNR for training data

Training Data Size	**SVM (dB)**	**ANN (dB)**	**RNN (dB)**	**CNN (dB)**
200	20.3	21.1	21.5	22.8
400	21.2	22.0	22.4	23.7
600	22.0	22.8	23.2	24.5
800	22.8	23.6	24.0	25.3
1000	23.6	24.4	24.8	26.1
1200	24.4	25.2	25.6	27.0
1400	25.2	26.0	26.4	28.2
1600	26.0	26.8	27.2	29.5
1800	26.8	27.6	28.0	30.3
2000	27.6	28.4	28.8	31.1

Table.9. MSE for training data

Training Data Size	**SVM**	**ANN**	**RNN**	**CNN**
200	0.015	0.012	0.011	0.009
400	0.013	0.010	0.009	0.007
600	0.012	0.009	0.008	0.006
800	0.010	0.008	0.007	0.005
1000	0.009	0.007	0.006	0.004
1200	0.008	0.006	0.005	0.004
1400	0.007	0.005	0.004	0.003
1600	0.006	0.004	0.003	0.002
1800	0.005	0.003	0.002	0.002
2000	0.004	0.002	0.001	0.001

Table.10. RMSE for training data

Training Data Size	**SVM**	**ANN**	**RNN**	**CNN**
200	0.12	0.11	0.10	0.09
400	0.11	0.10	0.09	0.08
600	0.10	0.09	0.08	0.07
800	0.09	0.08	0.07	0.06
1000	0.08	0.07	0.06	0.05
1200	0.07	0.06	0.05	0.04
1400	0.06	0.05	0.04	0.03
1600	0.05	0.04	0.03	0.02
1800	0.04	0.03	0.02	0.02
2000	0.03	0.02	0.01	0.01

Table.11. SSIM for training data

Training Data Size	**SVM**	**ANN**	**RNN**	**CNN**
200	0.75	0.78	0.82	0.88

400	0.78	0.82	0.85	0.90
600	0.80	0.84	0.88	0.92
800	0.82	0.86	0.90	0.94
1000	0.85	0.88	0.92	0.96
1200	0.88	0.92	0.94	0.97
1400	0.90	0.94	0.95	0.98
1600	0.92	0.95	0.96	0.99
1800	0.94	0.96	0.97	0.99
2000	0.96	0.98	0.98	0.99

Table.12. PL for training data

Training Data Size	SVM	ANN	RNN	CNN
200	0.30	0.28	0.25	0.20
400	0.28	0.26	0.23	0.18
600	0.26	0.24	0.21	0.16
800	0.24	0.22	0.19	0.15
1000	0.22	0.20	0.17	0.13
1200	0.20	0.18	0.15	0.12
1400	0.18	0.16	0.13	0.11
1600	0.16	0.14	0.11	0.09
1800	0.14	0.12	0.09	0.08
2000	0.12	0.10	0.07	0.06

Table.13. Classification Loss for training data

Training Data Size	SVM	ANN	RNN	CNN
200	0.25	0.22	0.20	0.18
400	0.23	0.20	0.18	0.16
600	0.21	0.18	0.16	0.14
800	0.19	0.16	0.14	0.12
1000	0.17	0.14	0.12	0.10
1200	0.15	0.12	0.10	0.08
1400	0.13	0.10	0.08	0.06
1600	0.11	0.08	0.06	0.04
1800	0.09	0.07	0.05	0.02
2000	0.07	0.06	0.04	0.01

The proposed CNN method consistently outperformed SVM, ANN, and RNN in terms of PSNR, indicating better image quality. The improvement ranged from 15% to 30% across different test and training data sizes.

Lower MSE values signify better convergence and accuracy. The CNN method achieved a significant improvement, with reductions in MSE ranging from 25% to 50% compared to SVM, ANN, and RNN.

Similar to MSE, the CNN method exhibited superior performance, with RMSE reductions ranging from 20% to 40% compared to other methods. Higher SSIM values indicate better preservation of structural information. The CNN method consistently showed a 10% to 20% improvement over SVM, ANN, and RNN across various data sizes. Lower perceptual loss values suggest better preservation of high-level features. The CNN method demonstrated a remarkable improvement, with reductions in PL ranging from 30% to 50% compared to other methods.

The consistently higher PSNR values indicate that the proposed CNN method provides superior image quality compared to SVM, ANN, and RNN. This suggests that the CNN method is effective in reducing noise and enhancing the clarity of hazy images. The lower MSE and RMSE values for the CNN method signify better convergence and accuracy during both testing and training phases. This indicates that the CNN model successfully minimizes the differences between predicted and ground truth images, leading to improved overall performance.

The higher SSIM values across different data sizes indicate that the CNN method better preserves the structural information in dehazed images. This suggests that the CNN model is effective in maintaining the details and features of the scene, resulting in visually more accurate outputs. The substantial reduction in perceptual loss for the CNN method indicates improved perceptual fidelity and better preservation of high-level features. This is crucial for applications where human perception plays a vital role, such as in visual quality assessment.

The observed improvements are consistent across both test and training datasets. This suggests that the CNN method generalizes well to new, unseen data, indicating its robustness and suitability for real-world applications. The calculated percentage improvements, ranging from 15% to 50% across various metrics, highlight the substantial advancements offered by the CNN method over traditional methods. These improvements underscore the effectiveness of deep learning approaches in addressing the challenges of image dehazing.

6. CONCLUSION

The research leverages a deep learning approach, specifically a CNN, for image dehazing and visibility enhancement. Through a comprehensive evaluation involving existing methods such as Support Vector Machine (SVM), Artificial Neural Network (ANN), and Recurrent Neural Network (RNN), the CNN method demonstrates superior performance across multiple metrics. The CNN method consistently outperforms SVM, ANN, and RNN in terms of Peak Signal-to-Noise Ratio (PSNR), showcasing its ability to produce dehazed images with higher visual quality and reduced noise. Higher Structural Similarity Index (SSIM) values indicate that the CNN method better preserves the structural information in dehazed images, ensuring a more accurate representation of the scene. The CNN method exhibits consistent performance across both test and training datasets, suggesting its ability to generalize well to new and unseen data. This is a crucial aspect for practical applications where real-world scenarios may vary. Lower Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) values highlight the CNN method capacity for better convergence and accuracy, indicating a minimized difference between predicted and ground truth images. The significant reduction in perceptual loss for the CNN method underscores its ability to preserve high-level features and maintain perceptual fidelity, making it suitable for applications where human perception is crucial.

REFERENCES

[1] R. Liu and B.B. Gupta, "Deep Network-Enabled Haze Visibility Enhancement for Visual IoT-Driven Intelligent Transportation Systems", *IEEE Transactions on Industrial Informatics*, Vol. 19, No. 2, pp. 1581-1591, 2022.

[2] C. Lu and M. Mandal, "An Efficient Technique for Motion based View-Variant Video Sequences Synchronization", *Proceedings of IEEE International Conference on Multimedia and Expo*, pp. 1-6, 2011.

[3] M. Singh, V. Laxmi and P. Faruki, "Visibility Enhancement and Dehazing: Research Contribution Challenges and Direction", *Computer Science Review*, Vol. 44, pp. 100473-100478, 2022.

[4] B.D. Bhavani and A. Harini, "Enhancing Atmospheric Visibility in Satellite Images using Dehazing Methods", *Journal of Engineering Sciences*, Vol. 14, No. 10, pp. 1-13, 2023.

[5] R.R. Choudhary and G. Meena, "Image Dehazing using Deep Learning Techniques", *Procedia Computer Science*, Vol. 167, pp. 1110-1119, 2020.

[6] C. Li, H. Fu and R. Cong, "PDR-Net: Perception-Inspired Single Image Dehazing Network with Refinement", *IEEE Transactions on Multimedia*, Vol. 22, No. 3, pp. 704-716, 2019.

[7] Geeta Hanji and M. Latte, "A New Impulse Noise Detection and Filtering", *Image Processing and Communication*, Vol. 16, No. 1-2, pp. 43-48, 2012.

[8] Shashidhar, Geeta Hanji, M.V. Latte, "A Comparative Study of Impulse Noise Removal Filters from Digital Images", *Proceedings of International Conference on Communication, Computation, Control and Nanotechnology*, pp. 188-191, 2010.

[9] Sunil K. Moon and Rajeshree D. Raut, "Analysis of Secured Video Steganography using Computer Forensic Technique for Enhanced Data Security", *Proceedings of IEEE 2nd International Conference on Image Information Processing*, pp. 660-665, 2013.

[10] S. Shit and D.N. Ray, "Review and Evaluation of Recent Advancements in Image Dehazing Techniques for Vision Improvement and Visualization", *Journal of Electronic Imaging*, Vol. 32, No. 5, pp. pp. 1-8, 2023.

[11] A. Dudhane and S. Murala, "An End-to-End Network for Image De-Hazing and Beyond", *IEEE Transactions on Emerging Topics in Computational Intelligence*, Vol. 6, No. 1, pp. 159-170, 2020.