2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF) | 979-8-3503-3436-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICECONF57129.2023.10083630

Monochromatic Image Dehazing Using Enhanced Feature Extraction Techniques in Deep Learning

Nisarg Doshi[1] Sagar Bhavsar[1] Rajeswari D[1]* and Srinivasan R[2]

[1]Department of Data Science and Business Systems, School of Computing, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur -603203
[2]Department of Computing Technologies, School of Computing, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur -603203
*(*Corresponding Author)*
E-mail : nd1894@srmist.edu.in sb7908@srmist.edu.in drajiit@gmail.com srinivar@srmist.edu.in

Abstract- Images photographed in foggy weather usually have poor visibility. To mitigate this problem researchers have come up with various image dehazing techniques. Now, more than ever, high-quality images that can be used to glean maximum information from autonomous systems are in high demand.This research work uses different Deep Learning (DL) architectures to draw out essential details from the picture and localize the information recovered to reduce the haze from the picture. The paper investigates to remove the hazes from the dehazed images using DL techniques. The first task of this proposed work attempts three pre-processing techniques namely, air light estimation, contextual regularization and boundary constraint. The second task of this work is to identify the suitable DL model to extract clear images from dehazed images. Evaluation metrics are PSNR value and SSIM value are used to estimate the values of dehazed images compared with clear images. Experimental results proves that AOD-Net outperforms good result with respect to PSNR value.

Keywords—CNN, boundaryconstraint, contextual regularization, Deep Learning, AOD-Net.

I.INTRODUCTION

A photograph taken in foggy conditions results in an image with low visibility. As depicted in Figure 1, the haze causes distant objects to lose contrast and blend in with their surroundings. The light reflected by these items is dimmed and diluted by the environment, and it interacts with light dispersed by various particles in the air before reaching the camera. Consequently, as these objects recede further from the camera, their colors fade and become more reminiscent of fog, with the degree of resemblance increasing with distance.

Dehazing is difficult because the intensity of the haze is reliant on the depth that is generally not known. Having only one fuzzy image to work with also makes the task more challenging because of the lack of input variety. Due to their reliance on either a large number of hazy photos as input or extra previous knowledge [1], most current dehazing technologies provide a reliance that is frequently unworkable in a wide range of real-world scenarios.

When utilized to single-image dehazing with more rigorous priors or assumptions, the atmospheric scattering model, which describes how an image originates when haze is present, has demonstrated significant progress. According to this theory, a fuzzy image is one where *S* is the discerned blurry image and *P* is the ambient light, Qis the transmission, representing the fraction of the luminance

Figure 1. The image above shows hazy image(top left) and dehazed image (top right), boundary constrain output (bottom left) and contextual regularization output (bottom right).

$$S(m) = P(m)Q(m) + R(1 - Y(m)) \ldots\ldots\ldots (1)$$

which is not dispersed and get to the camera, and R is the overall atmospheric light. Previous haze

removal methods, depends on image degradation technique, typically relied on either more detail knowledge or various inspections of the same scene. Samples of the same research are given in [2], [3]. Narasimhan and S.K.Nayar et al. [2] observe how the air light is partially polarized due to the atmospheric particles scattering some of it. From this result, a fast method for haze reduction using two pictures put across a polarizer at various angles is developed. The approach proposed in [4] uses quad tree searchto evaluate the transmission method and locate a region that is indicative of airlight scatter.

The linear model is utilized to estimate the transmission function, which is an approach that yields satisfactory results.

The proposal of [5] for the use of pre-processing techniques used to enhance important details in the hazy image can effectively make any model more efficient and effective is particularly interesting. We used the pre-processing techniques along with five traditional CNN models, namely, AlexNet, DenseNet, VGG16, ResNet is used by us to compare and contrast their effectiveness to get better results.

II. Literature Survey

Dehazing a single image is a difficult problem because less scene structure information is accessible. Tremendous progress has been made in recent years. Research into novel image models and priors provides valuable insight that aids in these developments.Fattal [6] propose a new model of image generation that incorporates surface shading and scene transmission. To calculate the scene transmission from a hazy image, it is necessary to assume that the two functions are statistically uncorrelated locally and then partition the image into sections with constant albedo. Tan [7] suggests increasing the local contrast of a fuzzy picture to make it more legible. Particularly in areas with exceptionally dense hazes, this approach can yield rather convincing results. However, the restored image often has significant haloing and distorted colors.

The fundamental finding of [8] is that maximal local patches in haze-free outdoor photographs has almost no pixels with little intensity in minimum one color channel, leading them to suggest a dark channel prior for monochromatic picture dehazing. When combined with a gentle mating procedure, the preceding can generate a convincing, haze-free, high-quality result. The model of a picture as a factorial Markov random domain proposed by L.Kratz and K.Nishino[9] treats scene albedo and depth as two exactly independent latent layers. A common presumption maximization approach is utilized to do the image factorization. This technique is capable of restoring a clear picture with crisp edges, although the end result is often too saturated.

The work of [10] can be is also fascinating. By constructing a CNN-based end-to-end network, they are able to train two distinct dehazing techniques on a haze-dense and haze light dataset, therefore improving their models' ability to detect and remove haze from images. For each degree of visibility present in the original image before dehazing, they select the associated model. This study compares and contrasts photographs that range in visibility in terms of luminance, haze, sharpness, contrast, intensity, and color characteristics. Additionally, a novel CNN-based hazy image categorization network with a minimal architecture has been developed.

The successful work of H.Zhang et. al's[11] GAN in creating natural-looking pictures has sparked substantial academic research into their potential. H.Zhang and V.M.Patel provide a combined discriminator depend on GAN to determine if the matched dehazed image and the evaluated transmission map are authentic in order to better include the mutual structural knowledge between the evaluated t and the dehazed outcome [12]. R.Mondal et al [13] demonstrate how to do multi-scale image dehazing utilizing a deep perceptual pyramid network, contemporary dense blocks, and residual blocks. With this technique, scene context is handled throughout the decoding process, due to an encoder-decoder design with a pyramid pooling phase. The proposed method using a bi-directional regularity loss to evaluate$y$ and Z in order to reconstruct J with a fully CNN and some fine-tuned adjustments. H. Yang et. al [14] explicitly describe the relationship between Y, Z, and U using a bilinear CNN, and then estimate $Y$and Z using the recommended bilinear composition loss function.

III. PROPOSED SYSTEM

A. Dataset Description

For model training and evaluation, we utilized the "REalistic Single Image DEhazing" (RESIDE) dataset. This huge benchmark comprises both artificial and real-world hazy pictures that may be used to test all known single-image dehazing approaches. An Outdoor Training Set is used for this project (OTS). A collection of blurry images captured in the real world that may be used to test and compare single-image dehazing techniques. The dataset is split into five subsets, each of which can be used for a different kind of training or evaluation, showcasing the variety of both the data sources and the image contents used. Both a "train" and "test" set of data were created. For the RESIDE OTS dataset, 80% of the images came from the train dataset, while the remaining 20% came from the test dataset. The dataset included about 13000 images, 350 of which were unique.

The dataset can be classified into two categories:

- Clear Images
- Hazy images

B. Pre Processing Techniques

Images are scaled, rearranged, and turned to arrays as part of the first processing of the dataset. Along with it the dataset also undergoes two other highly level pre-processing techniques which are specifically intended to be implemented to enhance the detail components necessary for effective image dehazing thus increasing the efficiency of the model.

Figure 2. The boundaries of the radiance cube. (Left) Blurry image. (Right) Pre-processed version made using the boundary constraint technique.

Figure 3. Perspective on Contextual Regularization using Weighted L1-norm. (left) original blurry image, (right) pre-processed version using Weighted L1-norm based Contextual Regularization.

1. Boundary Constraint

This method is used to develop the fog and dust image model and to estimate the transmittance function roughly. The optimum transmittance function, which is computed, is then used to restore the low illumination dust and haze image using the nonlinear context regularization approach based on logarithmic transformation. To calculate the multiple of the logarithmic transformation, the image's maximum luminance value is used. Figure 2 shows a hazy image with its corresponding image which is pre-processed using boundary constraint from radiance cube.

2. Contextual Regularization

Using the boundary constraint method, the haze and dust model is created roughly for estimating the transmittance function. The optimized transmittance function, which is calculated using the nonlinear context regularization technique based on logarithmic transformation, is then used to restore the low illumination fog and dust image. The multiple of the logarithmic transformation is calculated using the image's maximum luminance value.

In addition to the contextual regularization, a series of high-order filters is employed. In order to keep the image's contours intact, it uses a Laplacian operator and seven Kirsch operators. In Figure 3, we see a foggy image and its corresponding, contextual regularization-preprocessed counterpart.

C. Proposed Architecture

Figure 4 illustrates the method flow for the proposed method. The proposed system is designed as follows: A hazy image is taken as input. The image is not optimum to be directly given to the neural network for training, thus a set of preprocessing techniques are applied on the top of the hazy images

following which the pre-processed image is feed to the model for evaluation, effectively providing us with the dehazed image.

In the pre-processing module, essential features are extracted from the image such as amount of air light in the atmosphere using air light estimation, the edges and depth of the content in the image using contextual regularization and boundary constraint from radiance cube. With the help of feature extraction, the information contained in an unprocessed data set can be converted into quantifiable features for further analysis.

D. Convolutional Neural Network (CNN)

CNNs' main benefits lie in their ability to provide a dense network that efficiently makes predictions, identifies objects, etc. Multiple layers of a CNN (or ConvNet) can be trained to recognize various aspects of an image. Each image has a filter or kernel applied to it, and the results improve and become more detailed as you move through the layers. The filters can begin with basic features at the lower levels [15-16].

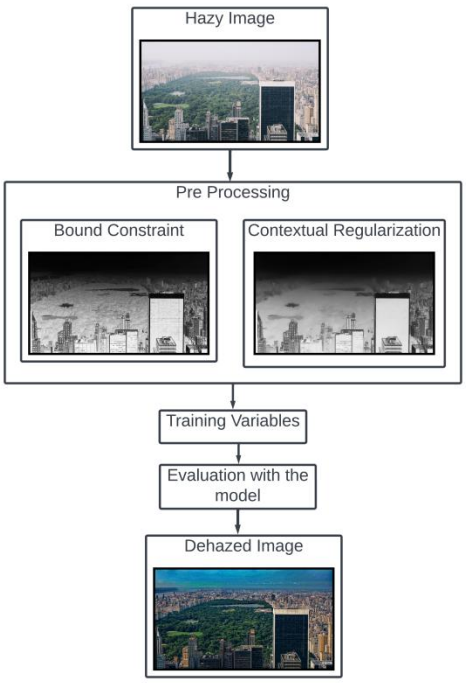

Figure 4. Proposed Architecture

The filters check and identify features that uniquely represent the input object at each layer by increasing in complexity. As a result, the partially recognized image produced by one layer of convolving is fed into the next. In the final layer, an FC layer, the CNN makes the identification of the image or object it represents.

E. VGG16

- The 16 in VGG16 indicates that there are 16 weighted layers. There are a total of 21 layers in VGG16 (13 convolutional, 5 Max Pooling, 3 Dense), but only 16 weight layers (the learnable parameters layer) [17].
- Tensor sizes of 224 and 244 with three RGB channels are accepted by VGG16 as input.
- VGG16 employs Multi-layer convolutional filters with a 3x3 design to stride 1 as well as the identical padding and the maxpool layer like a 2x2 filter to stride 2.
- There are an equal number of max pool and convolution layers. Conv-1 consists of 64 filters, Conv-2 of 128 filters, Conv-3 of 256 filters, and Convs 4 and 5 of 512 filters each.
- After the convolutional layers, there are three Fully-Connected layers. Initial two are equipped with 4096 channels both, whereas the third has 1000 channels and does 1000-way ILSVRC classification. After all the other layers were added, the soft-max layer is all that's left.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1_1 (Conv2D)            (None, 412, 548, 64)      1792

 conv1_2 (Conv2D)            (None, 412, 548, 64)      36928

 max_pooling2d_1 (MaxPooling  (None, 206, 274, 64)     0
 2D)

 conv2_1 (Conv2D)            (None, 206, 274, 128)     73856

 conv2_2 (Conv2D)            (None, 206, 274, 128)     147584

 max_pooling2d_2 (MaxPooling  (None, 103, 137, 128)    0
 2D)

 conv3_1 (Conv2D)            (None, 103, 137, 256)     295168

 conv3_2 (Conv2D)            (None, 103, 137, 256)     590080

 conv3_3 (Conv2D)            (None, 103, 137, 256)     590080

 max_pooling2d_3 (MaxPooling  (None, 51, 68, 256)      0
 2D)

 conv4_1 (Conv2D)            (None, 51, 68, 512)       1180160

 conv4_2 (Conv2D)            (None, 51, 68, 512)       2359808

 conv4_3 (Conv2D)            (None, 51, 68, 512)       2359808

 max_pooling2d_4 (MaxPooling  (None, 25, 34, 512)      0
 2D)

 conv5_1 (Conv2D)            (None, 25, 34, 512)       2359808

 conv5_2 (Conv2D)            (None, 25, 34, 512)       2359808

 conv5_3 (Conv2D)            (None, 25, 34, 512)       2359808

 max_pooling2d_5 (MaxPooling  (None, 12, 17, 512)      0
 2D)

 flatten (Flatten)           (None, 104448)            0

 fc_1 (Dense)                (None, 4096)              427823104

 dropout_1 (Dropout)         (None, 4096)              0

 fc_2 (Dense)                (None, 4096)              16781312

 dropout_4 (Dropout)         (None, 4096)              0

 output (Dense)              (None, 1000)              4097000

=================================================================
Total params: 463,416,104
Trainable params: 463,416,104
Non-trainable params: 0
_________________________________________________________________
```

Figure 5.1. Outline of VGG16Model

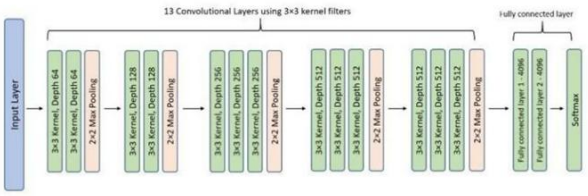

Figure 5.2 Architecture of VGG16

F. AlexNet

- AlexNet is a deep neural network that uses deep learning. The first convolutional network to take advantage of GPU acceleration.
- Maximum pooling is accomplished by means of the pooling layers.
- Since all of the layers are connected, the input size is fixed.
- The input size is typically given as 224 by 224 by 3, but in practice it is 227 by 227 by 3. This is likely due to padding during the input process.
- Sixty million parameters make up AlexNet's total size.

Layer	# filters / neurons	Filter size	Stride	Padding	Size of feature map	Activation function
Input	-	-	-	-	227 x 227 x 3	-
Conv 1	96	11 x 11	4	-	55 x 55 x 96	ReLU
Max Pool 1	-	3 x 3	2	-	27 x 27 x 96	-
Conv 2	256	5 x 5	1	2	27 x 27 x 256	ReLU
Max Pool 2	-	3 x 3	2	-	13 x 13 x 256	-
Conv 3	384	3 x 3	1	1	13 x 13 x 384	ReLU
Conv 4	384	3 x 3	1	1	13 x 13 x 384	ReLU
Conv 5	256	3 x 3	1	1	13 x 13 x 256	ReLU
Max Pool 3	-	3 x 3	2	-	6 x 6 x 256	-
Dropout 1	rate = 0.5	-	-	-	6 x 6 x 256	-

Figure 6.1 Outline and Architecture of AlexNet

G. DenseNet

A DenseNet is a CNN with dense connections between layers, as opposed to Dense Blocks, which link all layers directly. The feature maps of each layer are passed on as inputs to the layers below it, while the feature maps of the layers above it are ignored. DenseNet-121 has the following layers:

1. One 7x7 Convolution
2. Five 8 3x3 Convolution
3. Sixtyone - 1x1 Convolution
4. Four- AvgPool
5. One - Fully Connected Layer

H. ResNet

The ResNet architecture relies heavily on residual blocks. Convolutional layers, batch normalization layers, and nonlinear activation layers like ReLu are stacked in older architectures like VGG16. This technique can function with a relatively low number of convolutional layers—around 19 layers is the upper limit for VGG models. However, follow-up studies revealed that increasing the number of layers can significantly enhance CNN performance.

The ResNet architecture introduces the straightforward idea of connecting the final output of a chain of convolution blocks with an additional intermediate input. See an example of this down below.

```
Model: "sequential_111"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_222 (Conv2D)         (None, 410, 546, 6)       168

 average_pooling2d_222 (Aver  (None, 205, 273, 6)      0
 agePooling2D)

 conv2d_223 (Conv2D)         (None, 203, 271, 16)      880

 average_pooling2d_223 (Aver  (None, 101, 135, 16)     0
 agePooling2D)

 flatten_112 (Flatten)       (None, 218160)            0

 dense_323 (Dense)           (None, 120)               26179320

 dense_324 (Dense)           (None, 84)                10164

 output (Dense)              (None, 10)                850

=================================================================
Total params: 26,191,382
Trainable params: 26,191,382
Non-trainable params: 0
_________________________________________________________________
<keras.engine.functional.Functional at 0x7efccef95e10>
```

Figure 6.1 Outline of AlexNet

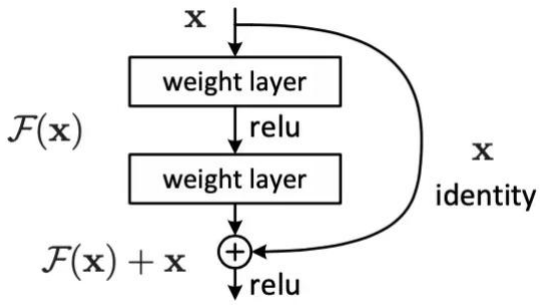

Figure 7.1 Residual Learning

IV. Experimental Results

Experimental results helps to investigate the preprocessing techniques with the DL model. Evaluation are carried out by Structural Similarity Index Measure (SSIM)and Peak-Signal-to-Noise Ratio (PSNR). This research work include 1300 images to evaluate the model. The dataset used in this research work are hazy images. Hazy images are extracted from RESIDE OTS (Outdoor training set). The model will dehazed the image by recognizing noise or haze from the essential details. The experiments are carried out in Google Colab Pro.

A. PSNR

The PSNR is the ratio of the strongest possible signal to the strongest possible noise that impairs the representation of a picture. The PSNR of an image can be calculated by comparing the perfect, noise-free of the image with dehazed of same image. Equation 2, helps to determine the PSNR value

$$PSNR = 20log_{10}(M - 1|RMSE) \ldots\ldots (2)$$

Where, M represents the highest possible intensity levels in an image. MSE is the mean squared error and RMSE is the abbreviation of Root Mean Squared Error.

B. SSIM

Digital images and videos of all kinds can have their perceived quality predicted using a technique called the SSIM, which is also used to measure structural similarity between two objects. Similarity between two images can be evaluated with the help of SSIM. To measure or predict image quality using the SSIM index, an original uncompressed or distortion-free image must be used as a baseline. Equation 3 determines the formula of SSIM as,

$$SSIM = \frac{(2.U_x.U_y+K_1)(2.S_{xy}+K_2)}{(U^2x.+U^2y+K_1)(S^2x.+S^2y+K_2)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

Where,

U_x = average x across a set of pixels;

U_y= mean y as measured by a set of pixels;

S^2x = the x-variance;

S^2y = dispersion of y;

$K_1 \& K_2$= variables used to make the division stable.

Figure 8 illustrates the full progression of a picture through the model's iterative processes. The figure depicts the haze image being sent to the boundary constraint pre-processing module, where the output, depicted in column 2, is obtained; the image is then sent to the following pre-processing module, where contextual regularization occurs, the Kirsch Filter is applied, and the morphological transformation of the image is performed. In the third column, we see the final product after various pre-processing techniques. The next stage is to feed the image into the neural network. Images produced by various CNN architectures are displayed in columns 4–7.The last column shows the how the image looks in actuality. The output from the modules should be substantially better or at least more in line with the real world situation.

In Table I, we can see the SSIM and PSNR for a number of different CNN model architectures. It is possible to gauge our model's efficacy with measurements like the SSIM and PSNR. The more accurate the dehazed image is, the higher the value of both metrics should be.

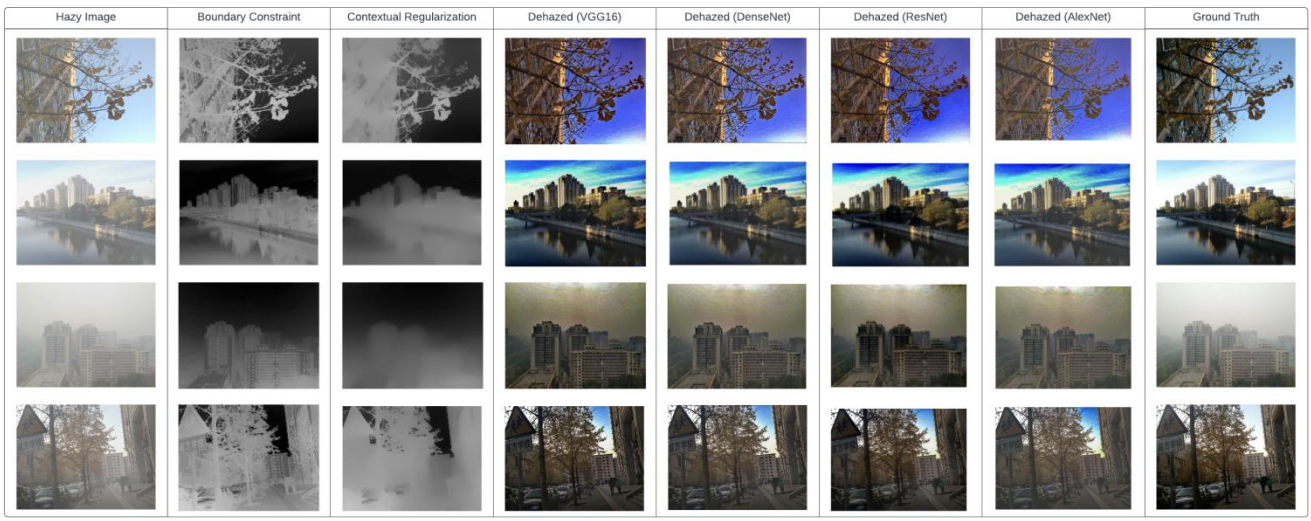

Figure 8. Complete life cycle of images in the model.

TABLE 1 PERFORMANCE COMPARISON ON THE OUTFOOR SOTS ON VARIOUS CNN ARCHITECTURES USED IN THIS STUDY

Deep Learning Method Used	***Average SSIM Value***	***Average PSNR Value***
VGG16	0.813	28.35
Alexnet	0.791	28.20
DenseNet	0.825	28.03
ResNet .	0.785	27.90

The database used to calculate the metrics values in Table II is the same one used to calculate the values in Table 1. (SOTS). A comparison of the performance of various existing methods are performed. The methods include: DehazeNet, MSCNN, AOD-Net, DCP.

TABLE 2 PERFORMANCE COMPARISON OF OUTDOOR SOTS USING PREEXISTING METHODS

	DCP	***Dehaze Net***	***MSCN N***	***AOD-Net***
PSNR (dB)	18.54	26.84	21.73	24.08
SSIM	0.710	0.826	0.831	0.873

V. CONCLUSION AND FUTURE WORK

In this study, multiple pre-processing strategies were employed to optimize the performance of the various CNN architectures tested. The results of the experiments demonstrated their potential for use in obtaining better results and the ability to feed more information to the neural network, thereby improving the model's accuracy. The average SSIM values show only moderate improvement when compared to the existing benchmark methods indicated in Table II, while the mean PSNR values of the various methods ranged from 27.90 in ResNet to 28.36 in VGG16, demonstrating significant strength. In future the technique also can be used with different set of images, in different environments like indoor images. Improved outcomes also can be achieved in the future by combining alternative pre-processing methods with other pre-existing models. The technique also can be used with different set of images, in different environments like indoor images.

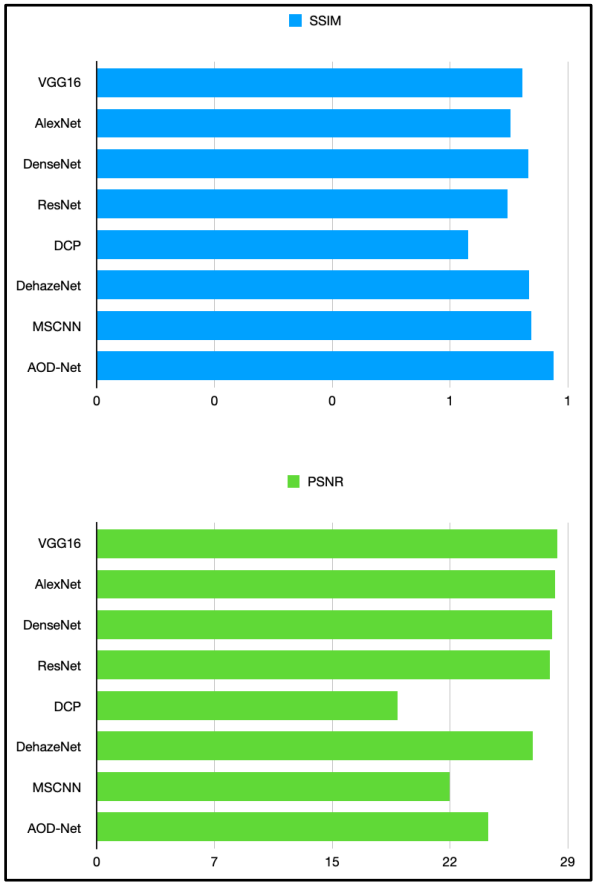

Figure 9. (Top) Comparison of various SSIM value (Bottom) Comparison of various PSNR values.

REFERENCES

[1] Y. Li, S. You, M. S. Brown, and R. T. Tan, "Haze visibility enhancement: A survey and quantitative benchmarking," Comput. Vis. Image Understanding, vol. 165, pp. 1–16, Dec. 2017.

[2] G. Narasimhan and S. K. Nayar. Vision and the atmosphere. IJCV, 48(3):233–254, 2002.

[3] Y. Y. Schechner, S. G. Narasimhan, and S. K. Nayar. Instant dehazing of images using polarization. In CVPR'01, volume 1, pages 325–332, 2001.

[4] W. Wang, X. Yuan, X. Wu, Y. Liu and S. Ghanbarzadeh, "An efficient method for image dehazing," 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 2241-2245, doi: 10.1109/ICIP.2016.7532757.

[5] J. Kopf, B. Neubert, B. Chen, M. Cohen, D. Cohen-Or, O. Deussen, M. Uyttendaele, and D. Lischinski. Deep photo: model-based photograph enhancement and viewing. In ACM SIGGRAPH Asia 2008, pages 116:1–116:10, 2008.

[6] R. Fattal. Single image dehazing. In ACM SIGGRAPH 2008, pages 72:1–72:9, 2008.

[7] R. T. Tan. Visibility in bad weather from a single image. In CVPR'08, pages 1–8, 2008.

[8] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. In CVPR'09, pages 1956–1963, 2009.

[9] L. Kratz and K. Nishino. Factorizing scene albedo and depth from a single foggy image. In ICCV'09, pages 1701–1708,Oct. 2009.

[10] X. Zhao, T. Zhang, W. Chen and W. Wu, "Image Dehazing Based on Haze Degree Classification," 2020 Chinese Automation Congress (CAC), 2020, pp. 4186-4191, doi: 10.1109/CAC51589.2020.9327091.

[11] H. Zhang, V. Sindagi, and V. M. Patel, “Multi-scale single image dehazing using perceptual pyramid deep network,”in Proc. IEEE Conf. Workshop on Comp. Vis. Patt. Recog., 2018, pp. 902–911.

[12] H. Zhang and V. M. Patel, “Densely connected pyramid dehazing network,” in Proc. IEEE Conf. on Comp. Vis. Patt. Recog., 2018, pp. 3194–3203.

[13] R. Mondal, S. Santra, and B. Chanda, “Image dehazing by joint estimation of transmittance and airlight using bidirectional consistency loss minimized fcn,” in Proc. IEEE Conf. Workshop on Comp. Vis. Patt. Recog., 2018, pp. 920– 928.

[14] H. Yang et al., “Image dehazing using bilinear composition loss function,” in arXiv preprint arXiv:1710.00279, 2017.

[15] M. Yadav, R. Goel and D. Rajeswari, "A Deep Learning Based Diabetic Retinopathy Detection from Retinal Images," 2021 International Conference on Intelligent Technologies (CONIT), 2021, pp. 1-5, doi: 10.1109/CONIT51480.2021.9498502.

[16] O. T. Khan and D. Rajeswari, "Brain Tumor detection Using Machine Learning and Deep Learning Approaches," 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), 2022, pp. 1-7, doi: 10.1109/ACCAI53970.2022.9752502.

[17] R. Mathur, T. Chintala and D. Rajeswari, "Identification of Illicit Activities & Scream Detection using Computer Vision & Deep Learning," 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), 2022, pp. 1243-1250, doi: 10.1109/ICICCS53718.2022.9787991.