Image Defogging Algorithm Based on Inception Mechanism

Jiahao Geng, Zhuang Miao, and Kezheng Lin(✉)

School of Computer Science and Technology, Harbin University of Science and Technology, Harbin, China

540228873@qq.com, link@hrbust.edu.cn

Abstract. In order to solve the problem that the existing defogging algorithms can't differentiate according to the characteristics of different regions of the fogged image, an Image Defogging Algorithm Based on Inception Mechanism is proposed(I-defog algorithm). The attention mechanism is added to the algorithm to adaptively assign weights to the features of different regions; It is more accurate and effective to use the module with perception mechanism to predict the global value. The predicted value, transmittance and foggy image are input into the atmospheric scattering model to get the defogging image, and the defogging image is input into the Markov discriminant (PatchGAN) to judge whether it is true or not. The results show that the algorithm achieves good defogging effect on both indoor and outdoor images, and improves the brightness and saturation of defogging images.

Keywords: Pattern recognition · Image defogging · Deep learning · Attention mechanism · PatchGAN

1 Introduction

The long-term rapid development of China's economy has accelerated industrialization and urbanization, and the frequency of haze weather across the country has increased significantly. In this low visibility haze environment, due to the influence of suspended water droplets and aerosols in the atmosphere, the saturation and contrast of images obtained by outdoor shooting equipment have decreased, and many important details have been lost, It is not conducive to the feature extraction of subsequent equipment, and increases the difficulty of image processing, which makes it difficult for various security measures such as monitoring system and target detection system to play a normal role. Therefore, the study of image defogging algorithm has great research significance, practical value, and great application prospects [1, 2]. How to effectively restore degraded image in fuzzy environment to clear image has become more and more important [3–5].

Traditional defogging algorithms can be divided into classical image enhancement algorithms [6]. The first kind of classical algorithms, such as histogram equalization [7, 8], Retinex algorithm [9, 10], wavelet transform [11], and so on. Although it was

S. Shi et al. (Eds.): 6GN 2021, LNICST 439, pp. 308–321, 2022.
https://doi.org/10.1007/978-3-031-04245-4_27

proposed earlier and implemented simply, it did not consider the real cause of fog generation. The second kind of image defogging algorithm based on image restoration is to estimate the physical quantities in the atmospheric scattering model by using prior knowledge, and then inverse and restore the defogging image, such as dark channel prior defogging algorithm (DCP) [12], nonlocal prior defogging algorithm [13], variational model defogging algorithm [14], image defogging algorithm including sky region [15], The methods of defogging in elliptic model [16], adaptive fog attenuation algorithm [17], color attenuation model (CAP) [18]. Due to the prior knowledge can not adapt to all the fog scenes, the image defogging algorithm based on prior has poor robustness.

Image defogging algorithm based on deep learning can obtain atmospheric light value, medium transmittance and other information related to image haze through convolution neural network, which has good robustness. For example, dehazenet (dehaze) proposed by Cai uses convolution neural network to directly estimate the transmittance and atmospheric light value of foggy image, so as to obtain foggy image [19, 20]; AOD net proposed by Li et al. Is an end-to-end defogging algorithm [21]; Zhang et al. Proposed a defogging network based on pyramid dense connection and u-net [22]; A domain adaptation paradigm proposed by Shao et al. Can make up the gap between synthetic domain and real domain [23].

This Algorithm Based on Inception Mechanism. In the algorithm, the attention mechanism can adaptively assign weights to the features of different regions of the image, which makes the algorithm more focused on the feature information related to the haze. The larger the haze is, the larger the weight will be, and the smaller the haze is, the smaller the weight will be; The intrusion mechanism can obtain more receptive fields, and reduce the computational complexity of the algorithm.

2 Related Work

2.1 Atmospheric Model

Atmospheric model is usually used to simulate the degradation process of foggy image. Because of its versatility, it is used in the research of image algorithm. The atmospheric model is shown in formula (1).

$$I(x) = J(x)t(x) + A(1 - t(x)) \tag{1}$$

In the formula: $I(x)$ is the observed value of foggy image at x, $J(x)$ is the observed value of the clear image at x, $t(x)$ used to represent the attenuation of light reaching the camera lens, From formula (1), we can know that as long as the atmospheric light value and perspective of foggy image are known, the foggy image can be restored to non foggy image, so the algorithm needs to correctly estimate the medium perspective and atmospheric light value. However, the accuracy of estimating the medium transmittance and atmospheric light value is uncertain, which will have a great impact on the defogging effect. The algorithm proposed in this paper adopts the mechanism of perception and attention, which can effectively improve the accuracy of the prediction of medium perspective and atmospheric light value.

2.2 Network Attention Mechanism

The traditional deep learning defogging network model can process the pixel features of each channel and each region equally. The corresponding weights of different regions with different haze should be different. In addition, each channel should also have different weight information, which should have different weights, It will waste resources on a large amount of useless information In order to make the algorithm adaptive weight allocation according to the characteristics of different regions, attention mechanism is added to the algorithm. The design of attention mechanism module is shown in Fig. 1.

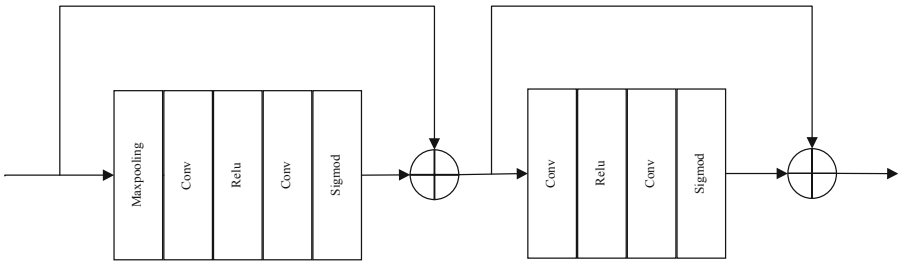

Fig. 1. Attention module

In the attention mechanism module, firstly, the channel spatial information is transformed into channel descriptors by maximizing pooling, as shown in formula (2).

$$g_C = Z_{\text{mp}}(F_C) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_C(i,j) \tag{2}$$

In the formula: $X_C(i,j)$ is the value of the image C channel at position (i,j), Z_{mp} is the maximum pooling function, The convolution kernel with the same size as the image is used to maximize pooling, convert the size of feature graph from $C \times H \times W$ to $C \times 1 \times 1$, Then, the feature map is input into the two-layer convolution layer, as well as the Relu activation function and Sigmod activation function, as shown in formula (3).

$$A_C = \sigma(\text{Conv}(\omega(\text{Conv}(g_C)))) \tag{3}$$

In the formula: ω is the Relu function, σ is the Sigmod function, Conv is the convolution function, A_C is the weight of the output, Finally, the corresponding F_C and A_C are multiplied, as shown in formula (4).

$$F_C^* = A_C F_C \tag{4}$$

F_C^* is output to the next stage, which is different from the first stage. In this stage, there is no pooling layer, and F_C^* is directly input into the two-layer convolution layer, as well as the Relu activation function and Sigmod activation function, as shown in formula (5).

$$A_{\text{p}} = \sigma\left(\text{Conv}\left(\omega\left(\text{Conv}\left(F_C^*\right)\right)\right)\right) \tag{5}$$

In the formula: ω is the relu function, σ is the Sigmod function, Conv is the convolution function, and A_p is the weight of the output. Finally, the output of the attention module is obtained by multiplying the corresponding F_C^* and A_p, as shown in formula (6).

$$F_p^* = A_p F_C^* \tag{6}$$

2.3 Perception Mechanism

Concept convolution neural network is proposed by googlenet. It has gone through four versions. Concept 1 uses 1 * 1, 3 * 3, 5 * 5 convolution kernels to extract features of different scales. At the same time, it also adds maximum pooling. In order to reduce the number of network parameters, reduce the difficulty of operation, and accelerate the convergence speed, In addition to batch normalization, two 3 * 3 convolution kernels are used to replace the original 5 * 5 convolution kernels. On the basis of concept 2, concept 3 proposes a parallel convolution structure, which makes convolution and pooling run in parallel, and then combines them to reduce the dimension without causing the loss of information extraction. On the basis of concept 3, concept 4 adds a residual network (RESNET), which achieves good results and improves the training speed. The concept module used in this article is shown in Fig. 2.

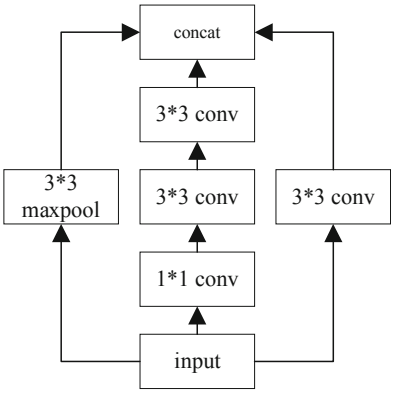

Fig. 2. Introduction mechanism

3 Method of This Paper

The foggy image is input into the defogging network to predict the value. The predicted value of atmospheric light, the medium transmittance and the foggy image. The foggy image is input into the Markov discriminator to judge whether it is true or not. The overall framework of algorithmic network is shown in Fig. 3.

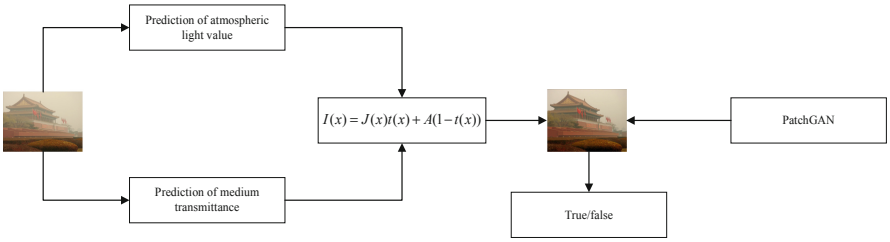

Fig. 3. Network framework

3.1 Prediction of Medium Transmittance

In addition, the attention mechanism can make use of the features with different weights to complement each other instead of focusing only on the local features. The media perspective estimation network based on attention mechanism used in this algorithm is shown in Fig. 4.

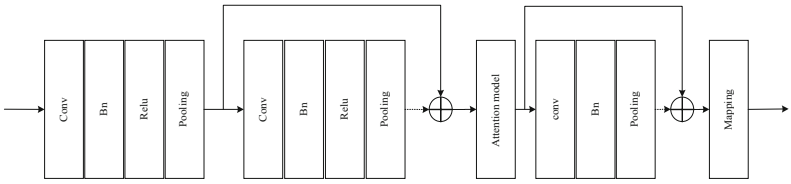

Fig. 4. Predictive transmission network

In order to obtain the haze, edge, texture and other information of different regions of the image for the subsequent use of deep network, the shallow network in the prediction transmittance module is used for feature extraction. Convolution layer and batch processing layer (BN layer) are used. In BN layer, the size of batch size can not be too large or too small. If the size of batch size is too small, the convergence speed of the model will be too slow, Appropriately increasing the batch size can increase the generalization ability and convergence speed of the model, but too large may lead to gradient disappearance or gradient explosion of the model. In this paper, the size of batch size is 64, and the relu activation layer and pooling layer are used to process the image, as shown in formula (7).

$$F_s = H_{SF}(I) \tag{7}$$

In the formula: I is the input foggy image, H_{SF} is the feature extraction function, F_s and is the output of shallow feature extraction. The F_s is input to the following residual module and attention module, and the F_s is down sampled. The residual module adds residual connection on the basis of feature extraction module. Residual connection can reduce the consumption of computing resources in the training of defogging network,

The subsequent attention mechanism module can automatically assign weights to the extracted features, so that the network module can focus more on processing the relevant feature information related to the medium transmittance, as shown in formula (8).

$$F_{\mathrm{d}} = H_{\mathrm{AM}}\left\{H_{\mathrm{RN,d}}\left\{\cdots\left\{H_{\mathrm{RN,1}}(F_{\mathrm{s}})\right\}\cdots\right\}\right\} \tag{8}$$

In the formula: $H_{\mathrm{RN,d}}$ represents the d residual convolution network function, H_{AM} represents the attention module function, and F_{d} represents the overall output of the function. In order to reconstruct the transmittance map, the deconvolution neural network is used for up sampling, as shown in formula (9).

$$F_{\mathrm{b}} = H_{\mathrm{DRN,b}}\left\{\cdots H_{\mathrm{DRN,2}}\left\{\left\{H_{\mathrm{DRN,1}}(F_{\mathrm{d}})\right\}\right\}\right\} \tag{9}$$

In the formula: $H_{DRN,b}$ represents the b residual convolution network, and F_b represents the overall output of the function. Finally, input F_b to the next layer of convolutional neural network, through the mapping layer, we can get the medium transmittance map, as shown in formula (10).

$$t = H_{\mathrm{MAP}}(F_{\mathrm{b}}) \tag{10}$$

3.2 Prediction of Atmospheric Light Value

To improve the convergence speed of the network and reduce the computational complexity in the network design of atmospheric light value prediction, we do not use the method of down sampling and up sampling, which is used to predict the medium perspective. Instead, we use the network based on the perception module to continuously down sampling. From the graph to the point, the final 1 * 1 result is the predicted atmospheric light value. The advantage of using the perception mechanism is that it can obtain more receptive fields, extract more global related information, and predict atmospheric light value more accurately, without the problem of taking the local optimal value as the global t value. At the same time, the perception module can speed up the network. The network with perception module is used to predict the atmospheric light value. The image processing process is shown in Fig. 5.

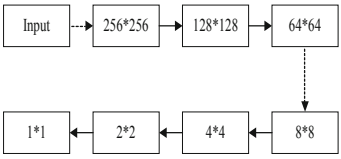

Fig. 5. Image processing process

3.3 Discriminator Network

In this paper, Markov discriminant (patchgan) is used to judge the true and false images after defogging. Different from the traditional discriminant model, in patchgan, the input is no longer a random high-frequency variable, but an image. When the image is input into patchgan, the discriminant will first segment the image into a n * n matrix (patch) to distinguish all patches, Finally, true or false is output according to the average value of all patch discrimination results of an image.

3.4 Algorithm Description

The first part is the mean square error loss function, as shown in formula (11).

$$L_{\mathrm{MSE}} = \frac{1}{N}\sum_{x=1}^{N} \|Y(x) - I(x)\|_2 \tag{11}$$

In the formula: $Y(x)$ represents the image before defogging, $I(x)$ represents the image after defogging, and N represents the total number of images contained in the training set. The second part is the discriminant loss function, as shown in formula (12).

$$L_{\mathrm{GAN}}(G, D) = E_{x,y}[\log_{10}^{D(x,y)}] + E_{x,y}[\log_{10}^{(1-D(x,G(x,z)))}] \tag{12}$$

In the formula: $E_{x,y}[\log_{10}^{D(x,y)}]$ is the probability to judge the original fog free image as true, $E_{x,y}[\log_{10}^{(1-D(x,G(x,z)))}]$ is the probability to judge the defogging image as false。

The expression of total loss function is shown in formula (13).

$$L = \partial L_{\mathrm{MSE}} + \beta L_{\mathrm{GAN}} \tag{13}$$

In the formula: ∂, β is the coefficient of the mean square error loss function and the discriminator error loss function respectively, and the default is 0.5.

This paper proposes a defogging algorithm, the specific process is shown in Algorithm 1.

algorithm 1 I-defog algorithm

input: Foggy image

output: Defogging image

(1) Input foggy imageI;

(2) Use formula (7) to extract shallow features and get feature map F_s;

(3) Firstly, F_s is downsampled by residual network to get , and attention mechanism is used to allocate weight $F_{\mathrm{d}} = H_{\mathrm{AM}}\left(F_s + \mathcal{F}\left(F_s, W_l\right)\right)$, H_{AM} is the whole function of attention mechanism, $\mathcal{F}\left(F_s, W_l\right)$ is the residual;;

(4) Then the deconvolution residual network is used for up sampling to get F_{b}, $F_{\mathrm{b}} = F_{\mathrm{d}} + \mathcal{F}\left(F_d, W_l\right)$;

(5) The transmittance map is obtained by using the mapping function, $t = H_{\mathrm{MAP}}\left(F_{\mathrm{b}}\right)$;

(6) Using Inception module, the atmospheric light value of foggy image is predicted $A = F_{\mathrm{Inc,n}}\left(...F_{\mathrm{Inc,1}}\left(I\right)\right)$;

(7) The defogging image J can be obtained, $J(x) = \dfrac{I(x) - A(1 - t(x))}{t(x)}$;

(8) Using PantchGAN to judge whether it is true or false;

(9) Further training the network, repeat (1) to (8) until the network loss is optimal, and the training is completed;

(10) Save the optimal model.

4 Experiments

4.1 Experimental Environment Configuration and Data Set

Hardware configuration: interi7-9700k processor (CPU), NVIDIA gtx-2060ti graphics card (GPU); The operating system is Ubuntu 20.04, and the model is built and trained under the deep learning framework pytorch.

The training data used in this paper is the reside data set, and the images in the data set are indoor foggy images, a total of 13900, which are synthesized from 1390 images in the existing indoor depth data set nyu2, each image is synthesized into 10 foggy blurred images. In addition, sots data set and HSTs data set will be used to verify algorithm.

4.2 Evaluating Indicator

PSNR first calculates the mean square error of two images. The smaller the mean square error is, the larger the PSNR value is, which indicates that the similarity of two images is greater, as shown in formula (14).

$$PSNR = 20\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \tag{14}$$

In the formula: MAX_I is the maximum value of pixel color, MSE is the mean square error of fogged image and non fogged image.

Different from PSNR, the more similar the image is and the smaller the image distortion is, as shown in formula (15).

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{15}$$

In the formula: μ_x represents the average of x, μ_y represents the average of y, σ_x^2 represents the variance of x,σ_y^2 represents the variance of y, σ_{xy} is the covariance of x and y, c_1 and c_2 are two constants.

4.3 Algorithm Validation

In this paper, RESIDE data set is used as training set, SOTS data set and HSTS data set are used as test set to verify the effectiveness of the algorithm. SOTS data set is composed of foggy images generated from indoor images, while HSTS data set is composed of foggy images generated from outdoor images.

The comparison results of some test sets between the algorithm in this paper and other algorithms, the comparison results of PSNR and SSIM in some sots datasets are shown in Fig. 6 and Fig. 7, and the comparison results of PSNR and SSIM in some HSTS datasets are shown in Fig. 8 and Fig. 9. It can be seen from Fig. 6, Fig. 7, Fig. 8 and Fig. 9 that when processing indoor foggy images, the algorithm in this paper performs well in both PSNR value and SSIM value, which is improved compared with cap algorithm, DCP algorithm and mscnn algorithm. The images before and after defogging with slightly lower evaluation index are analyzed, It is found that these images contain a large number of sky regions, which has an impact on the estimation of atmospheric light value, resulting in a slightly lower image evaluation index after defogging. The PSNR and SSIM values of cap algorithm and DCP algorithm decrease greatly when they deal with some outdoor images which are difficult to defog, and the defog effect is poor. Although the algorithm in this paper and dehaze algorithm decrease a little, the decrease is relatively stable.

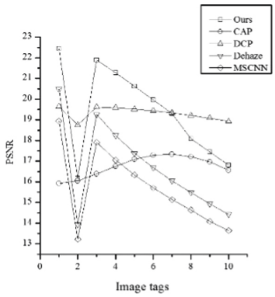

Fig. 6. Comparison of indoor PSNR

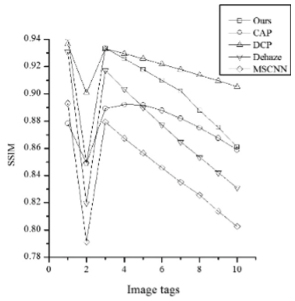

Fig. 7. Comparison of indoor SSIM

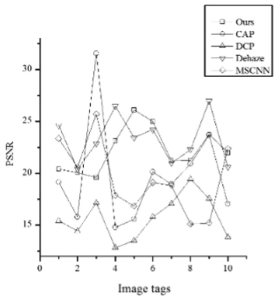

Fig. 8. Comparison of outdoor PSNR

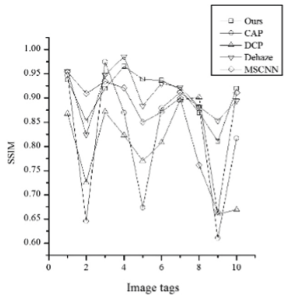

Fig. 9. Comparison of outdoor SSIM

The average comparison results of the algorithm in this paper and other algorithms in all test sets, In Table 1 and Table 2. SSIM is 0.024, 0.0059, 0.0008 and 0.0695 higher than cap, DCP, dehaze and MSCNN, and PSNR is 0.48 lower than dehaze, but 1.9, 1.89 and 3.74 higher than cap, DCP and MSCNN, respectively. On HSTs dataset, SSIM is 0.1079, 0.0843, 0.0052 and 0.0306 higher than cap, DCP, dehaze and MSCNN, respectively, and PSNR is 4.12, 6.37 and 2.75 higher than cap, DCP and MSCNN.

Table 1. SOTS dataset comparison results

Method	Evaluating indicator	
	SSIM	PSNR
CAP	0.8624	19.96
DCP	0.8505	19.97
Dehaze	0.8656	22.34
MSCNN	0.8169	18.12
Ours	0.8964	21.77

Table 2. HSTS dataset comparison results

Method	Evaluating indicator	
	SSIM	PSNR
CAP	0.7859	18.24
DCP	0.8095	15.99
Dehaze	0.8886	22.94
MSCNN	0.8632	19.61
Ours	0.8978	22.42

The visual comparison results of some outdoor scene images with fog in this algorithm and other comparison algorithms are shown in Fig. 10. It can be seen from Fig. 10 that the overall color of the image processed by cap algorithm is dark, and the sky part of the image is slightly distorted; DCP algorithm has serious distortion in sky area and character edge due to incomplete defogging. The values of four outdoor scene images are compared with other algorithms in this paper. The Fig. 11 and Fig. 12 show that the algorithm in this paper is relatively stable in terms of defogging performance. It performs well in PSNR and SSIM of four outdoor scene images, and some images have a large downward trend in DCP algorithm and MSCNN algorithm.

The values of four outdoor scene images are compared with other algorithms in this paper. The Fig. 11 and Fig. 12 show that the algorithm in this paper is relatively stable in terms of defogging performance. It performs well in PSNR and SSIM of four outdoor scene images, and some images have a large downward trend in DCP algorithm and MSCNN algorithm.

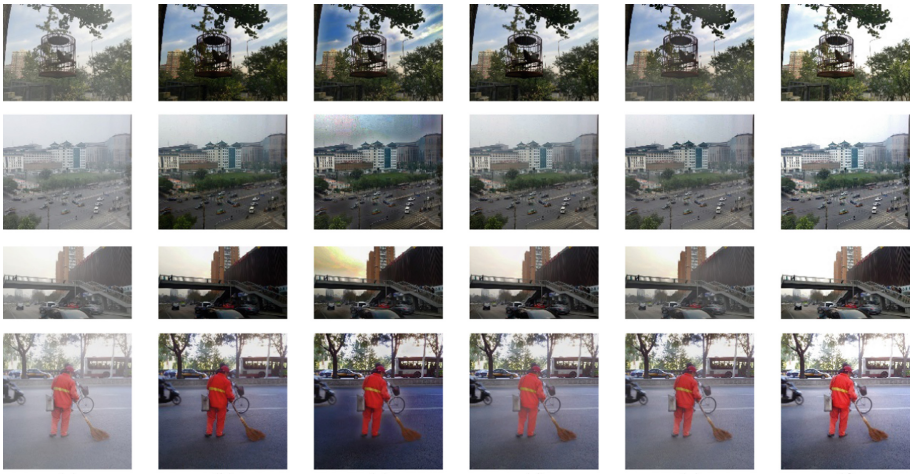

Fig. 10. Comparison of algorithms in outdoor (a) origin image (b) CAP results (c) DCP results (d) Dehaze results (e) MSCNN results (f) our results

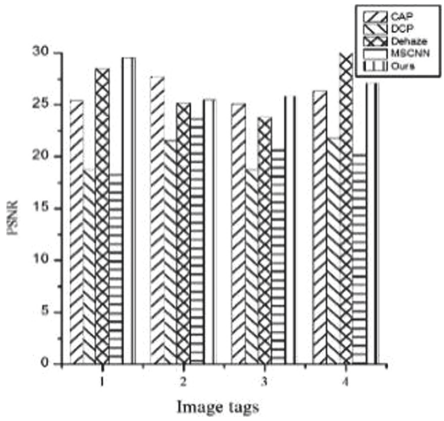

Fig. 11. Comparison of PSNR

4.4 Run Time Comparison

The average running time of this algorithm compared with cap algorithm, DCP algorithm, dehaze algorithm and MSCNN algorithm is shown in Table 3. The average processing time is 0.93 s. Compared with the comparison algorithm, the time is controlled within 1s, which has higher running efficiency.

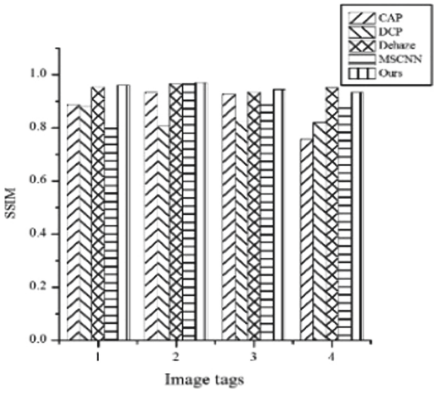

Fig. 12. Comparison of SSIM

Table 3. Comparison of average running time of different algorithms

Method	CAP	DCP	Dehaze	MSCNN	Ours
Time/s	1.43	10.11	1.77	1.71	0.97

5 Conclusions

This Algorithm Based on Inception Mechanism. On the one hand, it solves the problem that the traditional image defogging algorithm can not carry out differential processing according to different haze characteristics. On the other hand, the algorithm can make use of the features with different weights to complement each other instead of only focusing on the local features, makes better use of the global related information, makes the results more accurate. The results show that can achieves good defogging effect in both indoor and outdoor data sets, and improves the brightness and saturation of the image while effectively defogging.

References

1. Shu, Q.L., Wu, C.S., Zhong, Q.X., et al.: Alternating minimization algorithm for hybrid regularized variational image dehazing. Optik – Int. J. Light Electron Opt. **185**, 943–956 (2019)
2. Pu, H.F., Huang, Z.Y.: Review of image defogging algorithms. Softw. Eng. **24**(05), 2–6 (2021)
3. Tufail, Z., Khurshid, K., Salman, A., et al.: Improved dark channel prior for image defogging using RGB and YcbCr color space. IEEE Access **6**(1), 32576–32587 (2018)
4. Zhang, W.D., Dong, L.L., Pan, X.P., et al.: Single image defogging based on multi-channel convolutional MSRCR. IEEE Access **7**(1), 72492–72504 (2019)
5. Zhang, D.Y., Ju, M.Y., Qian, W.: Research status and Prospect of image defogging algorithm. J. Nanjing Univ. Posts Telecommun. (Nat. Sci. Ed.) **40**(05), 101–111 (2020)
6. Nolim, U.A.: Single image dehazing using adaptive dynamic stochastic resonance and wavelet-based fusion. Optik - Int. J. Light Electron Opt. **195**, 163111 (2019)

7. Gonzalez, R.C., Woods, R.E.: Digital image processing. Prentice Hall Int. **28**(4), 484–486 (2008)
8. Kim, T.K., Paik, J.K., Kang, B.S.: Contrast enhancement system using spatially adaptive histogram equalization with temporal filtering. IEEE Trans. Consum. Electron. **44**(1), 82–87 (1998)
9. Shi, R.X., Gao, B.L., Qiao, Y.J.: Improved Retinex low illumination image enhancement algorithm based on fusion strategy. Comput. Meas. Control **29**(04), 159–164 (2021)
10. Ou, J.M., Hu, X., Yang, J.X.: Low light image enhancement algorithm based on improved Retinex net. Pattern Recogn. Artif. Intell. **34**(01), 77–86 (2021)
11. Zhang, Z.H., Lu, J.G.: Fog image enhancement based on wavelet transform and improved Retniex. Comput. Appl. Softw. **38**(01), 227–231 (2021)
12. Li, Y.M., Zhang, X.J., Xie, B.W.: Improved dark channel prior image defogging algorithm based on brightness model fusion. Progr. Laser Optoelectron. **57**(22), 67–73 (2020)
13. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. IEEE Trans. Pattern Anal. Mach. Intell. **33**(12), 2341–2353 (2010)
14. Berman, D., Avidan, S.: Non-local image dehazing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1674–1682 (2016)
15. Guo, T., Li, N., Sun, L., et al.: A defogging method including sky region image. Progr. Laser Optoelectron. 1–14 [2021–06–16].http://kns.cnki.net/kcms/detail/31.1690.TN.20210409.0934.038.html
16. Jiao, Z., Fan, Z., Qian, L.: Research on fog removal based on RGB color space ellipsoid model. Progr. Laser Optoelectron. 1–15 [2021–06–16]. http://kns.cnki.net/kcms/detail/31.1690.TN.20210517.1011.002.html
17. Yan, Y., Zhang, J.: Image defogging algorithm based on fog concentration distribution and adaptive attenuation. Progr. Laser Optoelectron. 1–14 [2021-06-16]. http://kns.cnki.net/kcms/detail/31.1690.TN.20210301.1016.018.html
18. Liu, Y., Shang, J.X., Pan, L., et al.: A unified variational model for single image dehazing. IEEE Access **7**, 15722–15736 (2019)
19. Cai, B., Xu, X., Jia, K., et al.: Dehaze-net: an end-to-end system for single image haze removal. IEEE Trans. Image Process. **25**(11), 5187–5198 (2016)
20. Ren, W., Liu, S., Zhang, H., Pan, J., Cao, X., Yang, M.-H.: Single image dehazing via multi-scale convolutional neural networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 154–169. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_10
21. Li, B.Y., Peng, X.L., Wang, Z.Y., et al.: AOD-Net: all-in-all dehazing network. In: IEEE International Conference on Computer Vision, pp. 4770–4778 (2017)
22. Zhang, H., Patel, V.M.: Densely connected pyramid dehazing network. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, pp. 3194–3203 (2018)
23. Shao, Y., Li, L., Ren, W., et al.: Domain adaptation for image dehazing. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, pp. 2805–2814 (2020)