Single Image Snow Removal via Composition Generative Adversarial Networks

ABSTRACT:

Snowflakes attached to the camera lens can severely affect the visibility of the background scene and compromise the image quality. In this paper, we solve this problem by visually removing snowflakes to convert the snowy image into a clean one. The problem is troublesome; the information about the background of the occluded regions is completely lost for the most part. For removing snowflakes from a single image, we proposed a composition generative adversarial network. Different from the previous generative adversarial networks, our generator network comprises clean background module and a snow mask estimate module. The clean background module aims to generate a clear image from an input snowy image, and snow mask estimate module is used to produce the snow mask in an input image. During the training step, we put forward a composition loss between the input snowy image and composition of the generated clean image and estimated snow mask. We use a dataset named Snow100K[2] including indoor and outdoor scenes to train and test the proposed method. The extensive experiments on both synthetic and real-world images show that our network has a good effect and it is superior to the other state-of-the-art methods.

Bad weather, such as snowfall, can seriously decrease the quality of images and pose great challenges to computer vision algorithms. In view of the negative effect of snowfall, this paper presents a single-image snow removal method based on a generative adversarial network (GAN). Unlike previous GANs, our GAN includes an attention mechanism in the generator component. By injecting attention information, the network can pay increased attention to areas covered by snow and improve its capability to perform local repairs. At the same time, we improve the traditional U-Net network by combining it with the residual network to enhance the effect of the model when removing snowflakes from a single image. Our experiments on both synthetic and real-word images show that our method produces better results than those of other state-of-the-art methods.

CHAPTER-1
INTRODUCTION

With the progress in artificial intelligence, high-tech applications for various outdoor scenarios continue to emerge, such as automated driving systems and traffic monitoring. In most cases, the input images fed into these applications handling image-related tasks need to be sharp and clean so that the correct information can be extracted and processed. In practice, however, interferences are difficult to avoid, and they lead to degradation of the acquired images, poor visual effect, and less useful information in the images. Such images do not lead to satisfactory results after processing. Among various weather interferences, fog, rain, and snow are the most common. The study of image enhancement is therefore valuable for applications used for outdoor scene images. Existing research on image enhancement mainly falls into two types: model based and machine learning based. The first type mainly uses traditional models to characterize texture and background image separately and transforms the entire process into an iterative procedure through an appropriate optimization algorithm to obtain solutions. The latter type constructs a network architecture for image enhancement.

Kang et al. [1] proposed the use of a bilateral filter to decompose an image into high frequency and low frequency components. The low-frequency component essentially retains most information concerning the background image, while the high-frequency component mainly contains the texture information and rain streaks. The histogram of oriented gradient (HOG) feature descriptors are applied on the high frequency component. Dictionary-learning-based sparse coding is then used to extract the texture information of the background image from the high-frequency component, which, together with the low-frequency component, provides a final estimation of the background image.

The idea behind this method, though intriguing, yields ambiguous results. Chen and Hsu [2] suggested the separation of rain streaks and background image from a rain image via discriminative sparse coding. Experiments demonstrated that clear estimated background images were obtained, but the rain streak removal effect was less than satisfactory. As research advances and the understanding of actual rainy images deepens, researchers have found the formation of fog in different degrees along with the rain; this also affects the visual effect of the estimated background image. Based on patch-based learning, Li et al. [3] used Gaussian mixture models to characterize the priors of background and rain layers. The validity of this method in actual rain removal was proven in their study concerning the joint use of the defogging algorithm. By far, this method produces the best result among all the models described, but some missing details from the estimated background image leads to image blurring.

Compared to the application of models in image enhancement, the research on deep learning architecture for image enhancement has just emerged in recent years. Fu et al. [4] proposed a network architecture called DerainNet based on a deep convolutional neural network. In addition, considering the effect of fog caused by the rain, they suggested the performance of defogging after rain removal. In addition, Fu et al. simplified the learning mode of deep residual networks and proposed a deep detailed network to tackle deraining. The processing capacity of this network architecture was validated by use in denoising and

other processes. Yang et al. [5] suggested the detection of rain streak locations in rain images and varied the degrees of removal according to different intensities of the rain streaks. A multi-task deep network is, in this case, designed to carry out the detection and removal iteratively. They [6] also took into account the impact of fog on visual perception and attempted to employ a joint deraining-defogging-deraining methodology to eliminate rain and fog.

Experiments indicated that their method is particularly effective for heavy rain. In these image enhancement tasks, factors such as the varied size and shape of snowflakes and irregular trajectory make snow removal from the image even more challenging. Liu et al. [7] designed DesnowNet to deal with this problem, and it has proved to be effective experimentally. In order to solve the snow removal problem, this paper proposes a novel method - composition generative adversarial networks (CGAN), which can effectively remove the snowflakes in the image under different snow particle sizes (see Figure 1.1).

FIGURE 1.1. Sample results of our method. From left to right: Real world snowy images, pristine images.

The contribution of this paper are as follows:

1. This paper proposed a novel snow removal algorithm based on generative adversarial networks (GAN) called CGAN- (composite generative adversarial networks).The CGAN include generator network and discriminator network.
2. In the generator network, we use the new structure based on encoder-decoder architecture, it consists of clean background module and snow mask estimate module.
3. We develop an up-to-date loss function based on least squares loss, mean absolute error (MAE) loss and composition loss.

CHAPTER-2
BACKGROUND RESEARCH METHODOLOGIES

In this section, we briefly review the existing deep-learning based single image snow removal algorithms, and the development of GAN.

2.1. Single Image Snow Removal:

Single image snow removal is used to restore a clear image from a snowy image which is corrupted by snowflakes. This relationship can be formulated by

$$I = J \odot m + A \odot (1 - m)$$

where I represents snowy images,
A is the clear image,
m is the snow masks,
J is the chromatic aberration map, and
$\odot$ denotes element-wise multiplication.

Due to the success of convolutional neural networks (CNN) in classification and recognition [8]–[10], it has been applied in the snow removal. Liu et al. [7] proposed a network architecture called DesnowNet based on a multistage convolutional neural network. It first estimates the snow mask and aberration map by CNN, then uses a conventional method to recover clear images. However, DesnowNet has poor robustness to real world snowy images. Thus, we present a novel method for snow removal.

2.2. Generative Adversarial Network:

To generate realistic-looking images, in 2014, Goodfellow et al. [11] proposed the GAN framework. GAN consists of a generator network and a discriminator network. VOLUME 7, 2019 25017 Z. Li et al.: Single Image Snow Removal via CGANs.

The generator network is used to generate image from input image and the discriminator network aims to distinguish whether the generated image is from real data or the generator network. However, GAN is not stable in the training process and often produces artifacts such as noise and color shift in the generated images [12]. Arjovsky et al. [12] discussed the difficulties in GAN training caused by JS divergence approximation and proposed to use the Earth-Mover's (also called Wasserstein-1) distance W(q, p).

Furthermore, Mao et al. [13] proposed the least squares generative adversarial networks (LSGANs) which adopt the least squares loss function for the discriminator to overcome mode collapse, vanishing gradient, etc. in the training process. Recently, GAN has received increased attention. It has been applied to different image-to-image translation problems, such as style transfer [14], [15], photo enhancement [16], super resolution [17], and others [18]–[20].

CHAPTER-3
PROPOSED METHOD

In this section, we introduce the network architecture of the CGAN including the generator and the discriminator. In the following, we present loss function in details and show the results of the snow mask estimate module.

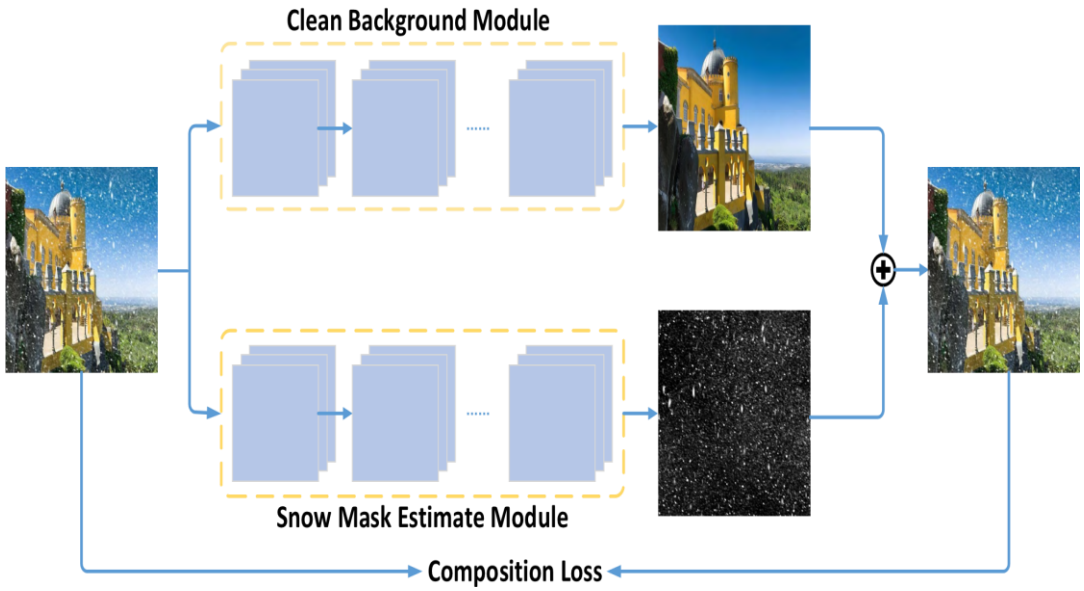

FIGURE 3.1. Architecture of our proposed generator network. It consists of a clean background module and a snow mask estimate module.

3.1. Generator Network:

The generator network architecture is illustrated in Figure 3.1. It consists of two modules, the clean background module and the snow mask estimate module.

The purpose of the clean background module is to learn the clean image from input snowy image, and snow mask estimate module aims to generate the snow mask from input snowy image, then use the constructed snowy image as the self-supervised information to guide the backpropagation.

3.1.1. Clean Background Module:

The clean background module is used to generate a clear image from an input snowy image. The proposed generator network as shown in Table 3.1 contains an encoding process and a decoding process. The encoding process is mainly based on the down-sampling operations and the decoding process mainly uses the up-sampling operations.

TABLE 3.1. Architecture of the clean background module and parameter setting. ‘‘conv’’ denotes the convolution, ‘‘uconv’’ denotes the deconvolution

Encoding	**Layer**	**Conv**	**Conv**	**Conv**	**Conv**	**Conv**	**Conv**	**Conv**	**Conv**	**Conv**	**Conv**
	Kernel	5x5	3x3	4x4	4x4	4x4	4x4	4x4	4x4	4x4	4x4
	Stride	1x1	1x1	2x2	2x2	2x2	2x2	2x2	2x2	2x2	2x2
	Channel	64	64	64	128	256	512	1024	1024	1024	1024
Decoding	**Layer**	**Uconv**	**Uconv**	**Uconv**	**Uconv**	**Uconv**	**Uconv**	**Uconv**	**Uconv**	**Uconv**	**Tanh**
	Kernel	4x4	4x4	4x4	4x4	4x4	4x4	4x4	4x4	3x3	-
	Stride	2x2	2x2	2x2	2x2	2x2	2x2	2x2	2x2	1x1	-
	Channel	1024	1024	1024	512	256	128	64	64	3	-

3.1.2. Snow Mask Estimate Module:

The snow mask estimate module is a pre-trained model and it is used to generate snow mask from an input snowy image. The proposed snow mask estimate module as shown in Figure 3.2 contains five strided convolution blocks, nine residual blocks (ResBlock) [21] and four transposed convolution blocks. Each ResBlock consists of a convolution layer, an instance normalization layer [22], and a ReLU activation layer [23]. Dropout [24] with probability of 0.5 is added to the generator after the first convolution layer in each ResBlock [21].

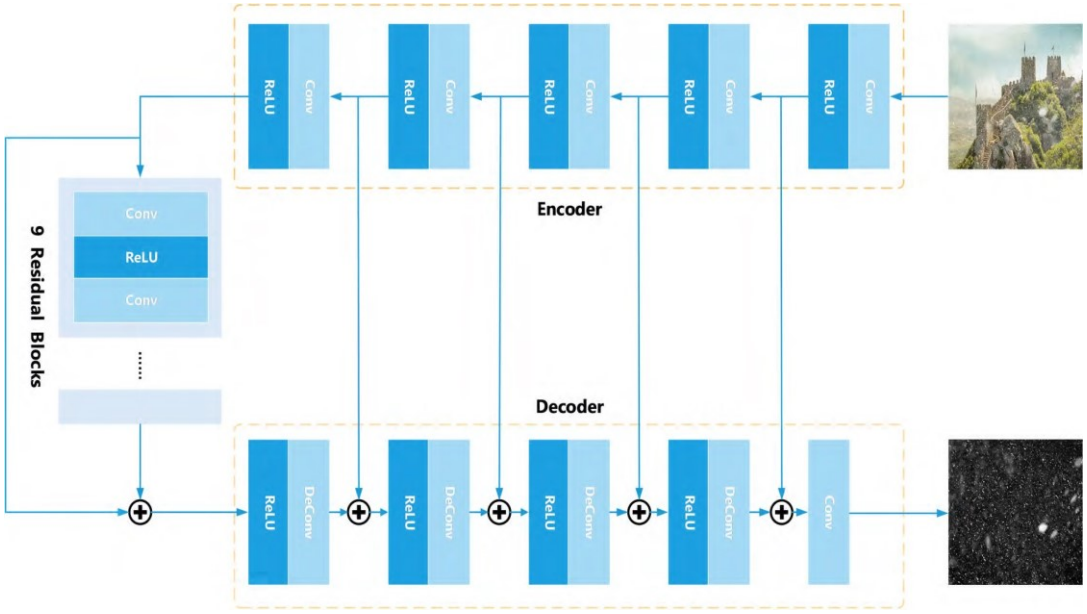

FIGURE 3.2.. Architecture of the snow mask estimate module.

The sample results of the snow mask estimate module as shown in Figure 3.3. The loss function of pre-trained snow mask estimate module can be written as:

$$L_E = \frac{1}{N}\sum_{i=1}^{N} \|z - M(x)\|_2^2$$

where N stand for the number of training images, x is the input snowy image, z is the corresponding snow mask and function ***M*(x)** is used to generate snow mask.

FIGURE 3.3. Example results of the snow mask estimate module. From top to bottom: Input snowy images, generated snow masks, ground truth.

3.2. Discriminator Network:

The discriminator network is used to classify whether a pair of images is real (1) or fake (0). The proposed discriminator network as shown in Figure 3.4 contains a heap of convolutional layers, where each convolutional layer is followed by a batch normalization layer and LeakyReLU activation layer. The last layer of the discriminator network is a sigmoid function, which outputs the probability of the input image pair to be real or fake.

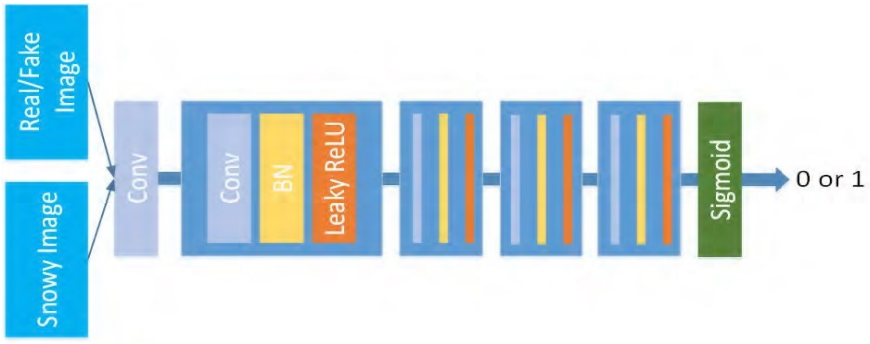

FIGURE 3.4. The architecture of the proposed discriminator network.

3.3. LOSS FUNCTION:

We formulate the loss function as the weighted sum of least squares loss [13], MAE loss and composition loss as:

$$\mathbf{L} = \alpha \mathbf{L_{ls}} + \beta \mathbf{l_{MAE}} + \Upsilon \mathbf{L_C}$$

3.3.1. Least Squares Loss (L_{ls}):

Most of the papers related to GANs, use vanilla GANs objective as a loss. However, we find that the CGAN using this loss is not able to remove the snow well and also product some artifacts and color shift on generated snowy images. As shown in the following, the quantitative results (Table 3) indicate that only using vanilla GANs objective does not generate clear images. Therefore, we use least squares loss as the critic function. the loss is calculated as the following:

$$L_{ls} = \mathbb{E}_{x,y \sim P_{data(x,y)}} \left[(D(x,y) - b)^2 \right] + \mathbb{E}_{x \sim P_{data(x)}} \left[(D(x, C(x)) - a)^2 \right]$$

where **x** is the input snowy image, **y** denotes the ground truth snow-free image, **C(x)** is the function of generate clean image and ***D*( )** is the function of discriminator, the setting of a and b which represents the labels indicating whether the sample is real will be described in chapter 4.

3.3.2. MAE Loss (L_{MAE}):

The benefit of using MAE loss is that it encourages the clean background module to generate images which are closer to ground truth images. The MAE loss between ground-truth snow-free image ***y*** and the generated snow-free image ***C*(*x*)** is given by:

$$L_{MAE} = \frac{1}{N} \sum_{i=1}^{N} \| y - C(x) \|_1$$

3.3.3. Composition Loss (L_C):

In order to remove the artifacts and color shift of the generated snowy image, we introduce composition loss helps to minimize pixel-level differences between input snowy image ***x*** and composition of the generated clean image ***C*(*x*)** and estimated snow mask ***M*(*x*)**, which is defined as:

$$L_C = \frac{1}{N} \sum_{i=1}^{N} \| x - (C(x) + M(x)) \|_1$$

CHAPTER-4
WORKING

In this section, we first introduce the dataset and training details of our method. In the following, this paper further analyzes and discuss the effect of the proposed algorithm including the discriminator architecture and loss function. Finally, we discuss the application of the proposed method.

4.1. Dataset:

In this study, a large-scale dataset named Snow100K[2] [7] was used for testing and training.

It consisted of

1) 100,000 synthesized snowy images,
2) the corresponding snow-free images, and
3) snow masks.

The snow-free images were captured using Flickr API [25]. To synthesize snow images, Liu et al. [7] use Photoshop to create 5,800 masks. Each mask contained snowflakes of different shapes. Corresponding examples are shown in Figure 4.1 and the number of training and testing set introduced in Table4.4.1.

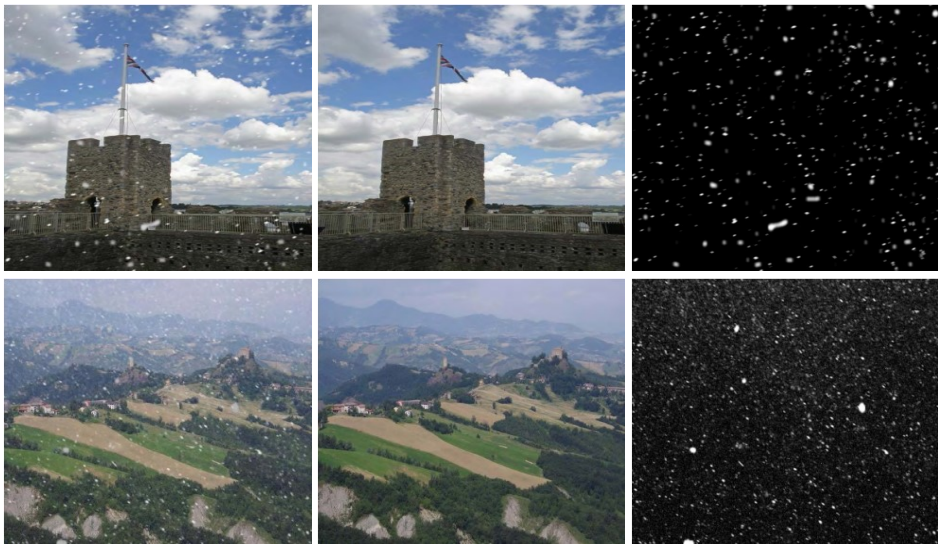

FIGURE 4.1. Sample image of Snow100K2.
From top to bottom: Sample synthetic snowy images, ground truth and corresponding snow masks.

TABLE 4.1. Number of each subset of Snow100K2 dataset.

SUBSET	Snow100K-S	Snow100K-M	Snow100K-L
Training	16643	16622	16735
Testing	16611	16588	16801

4.2. Training Details:

We train the network on an NVIDIA 1080Ti GPU. The proposed method is implemented using the TensorFlow 1.10.0 [26] and Python 3.6.0. Each layer of the proposed generator network consists of a convolution, instance normalization [22] and LeakyReLU.

All the training images are resized to 256 × 256. We set the parameters of batch-size, initial learning rate for Adam, and momentum to 1, 0.0002, and 0.5, respectively. Training stage stops at 200 epochs.

The parameters are initialized as follows: $\alpha = 1$, $\beta = 50$, $\gamma = 50$, $a = 1$, $b = 0$. As same as vanilla GAN, the generator network and the discriminator network are alternately updated.

4.3. Effect of the Loss Function In CGAN:

Introduction through CHAPTER 3, we advance a new loss function, which consists of least squares loss, MAE loss and composition loss. In the following, this paper will demonstrate the validity of this loss function through experiments.

1. **Adversarial Loss Vs. Least Squares Loss**:

   In order to confirm the effectiveness of least squares loss, we train the least squares loss [13] and adversarial loss [11] under the same network and use average PSNR and SSIM to evaluate the results separately. It can be seen in Table 2 that the least squares loss function gets higher PSNR value. Therefore, the proposed algorithm uses least squares loss.

2. **Effect of the Composition Loss:**

   In order to verify the effectiveness of our loss function, we experimented different loss functions under the same condition. It can be seen in Table 4.2 that the proposed loss function generates better results. We note that the method with Lls + Lc loss generates the results with higher PSNR values compared with the method with Lls + LMAE loss. The results from the second column and the third column show that LMAE helps to improve the SSIM value, and indicate that it is able to preserve the structures of images. Figure 4.2 shows some snow removal results examples with different loss functions and corresponding PSNR/SSIM. The method with the proposed loss function generates better results.

TABLE 4.2. Quantitatively evaluate the effect of the different loss functions in the proposed method.

LOSS	L_{ad}	L_{ls}	$L_{ls}+L_{MAE}$	$L_{ls}+L_c$	$L_{ls}+L_{MAE}+L_c$
PSNR	27.4523	28.2513	28.4634	29.3421	30.3972
SSIM	0.9067	0.9132	0.917	0.9232	0.9355

FIGURE 4.2. The effect of the proposed network with different loss function.
(c) Lad loss. (d) Lls loss. (e) Lls + LMAE loss. (f) Lls + Lc loss (g) Lls + LMAE + Lc loss.

4.4. Different Architecture Of Discriminator Network:

For purpose of prove the effectiveness of our discriminator network, we experimented and evaluated different architecture of the discriminator. The experimental settings of the discriminator are listed below:

a) **1 × 1 Pixel GAN discriminator:**
The 1 × 1 Pixel GAN discriminator structure is: C64 – C128. In this case, each layer is 1 × 1 filter size.
b) **16 × 16 Patch GAN discriminator:**
The 16 × 16 Patch GAN discriminator architecture is: C64 – C128.
c) **70 × 70 Patch GAN discriminator:**
The 70 × 70 Patch GAN discriminator architecture is: C64 – C128 – C256 – C512.
d) **286 × 286 Image GAN discriminator:**
In this special case, input image first resized to 286 × 286. The 286 × 286 Image GAN discriminator architecture is: C64 – C128 – C256 – C512 – C512 – C512,

where C denotes the output channel.

Figure 4.3 shows three real snowy images and the corresponding snow removal results generated by CGAN with different architecture of discriminator network, 70 × 70 PatchGAN discriminator generates better results. The other is able to remove some snowflakes, but the recovered images still contain snow partials and some artifacts. Therefore, the proposed discriminator network use 70 × 70 PatchGAN architecture.

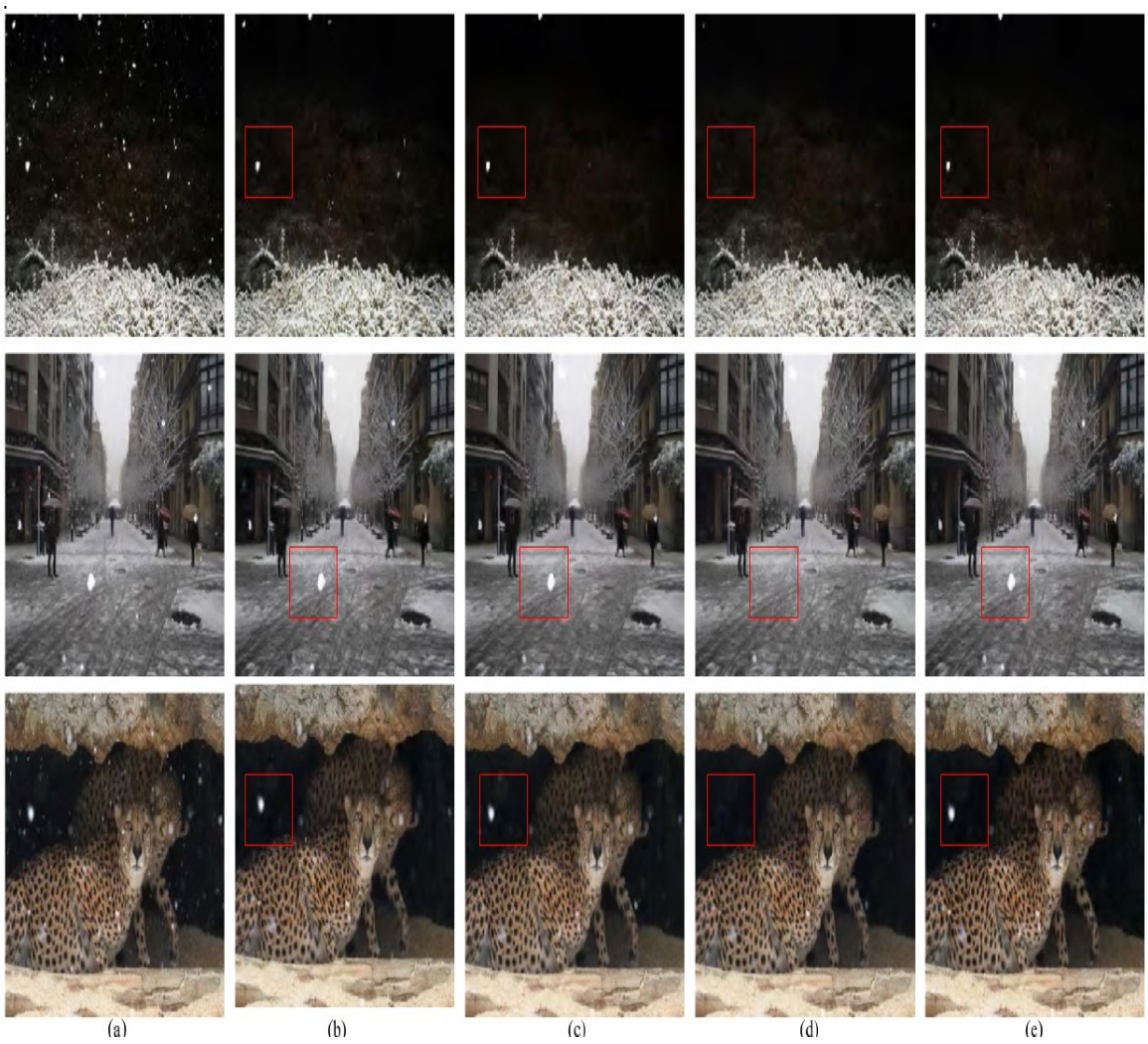

Figure 4.3: The effect of the different architecture of discriminator network
a)input images, b)1x1, c)16x16, d)70x70, e)286x286.

CHAPTER-5
APPLICATON OF METHODOLOGY

To provide further evidence that our image enhancement could be beneficial for computer vision recognition, we employ YOLO-V3 [27] to test our method. The results are shown in Figure 5.1. As can be seen, using our output image, the general recognition is better than the one without our image enhancement process.

FIGURE 5.1. Sample result of YOLO-V3. From left to right: Real world snowy image, generated clean image.

5.1. Comparison With Other Methods:

In this section, we compare our approach with the other state-of-the-art methods on synthetic dataset and real-world snowy images.

5.1.1. Quantitative Analysis:

We compare quantitative performance of different methods on the test images from the synthetic snowy image dataset, Snow100K[2] . Quantitative results corresponding to different methods are tabulated in Table 5.1. It can be clearly observed that the proposed method is able to achieve superior quantitative performance.

TABLE 5.1. Respective performances of the state-of-the-art method evaluated via Snow100K2's test set.

SUBSET	SNOW100K-S		SNOW100K-M		SNOW100K-L		OVERALL	
METRIC	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DESNOWNET	32.33331	0.95	30.8682	0.9409	27.1699	0.8983	30.1121	0.9296
OURS	30.4326	0.9612	31.2103	0.9431	29.5487	0.9021	30.3972	0.9355

5.1.2. Real Image Snow Removal:

The snow removal effect of the proposed method was evaluated using five real-world snow scene images. The removal outcomes are presented in Figure 5.2. The snow removal results of DesnowNet [7] show

that the first and second images were over-processed. In the second image, the distant lights were removed as snowflakes by DesnowNet. However, snow was not completely removed in the other three images. Our method led to clearer visual representations than DesnowNet did. As seen in the first image, our method yielded more refined processing results.

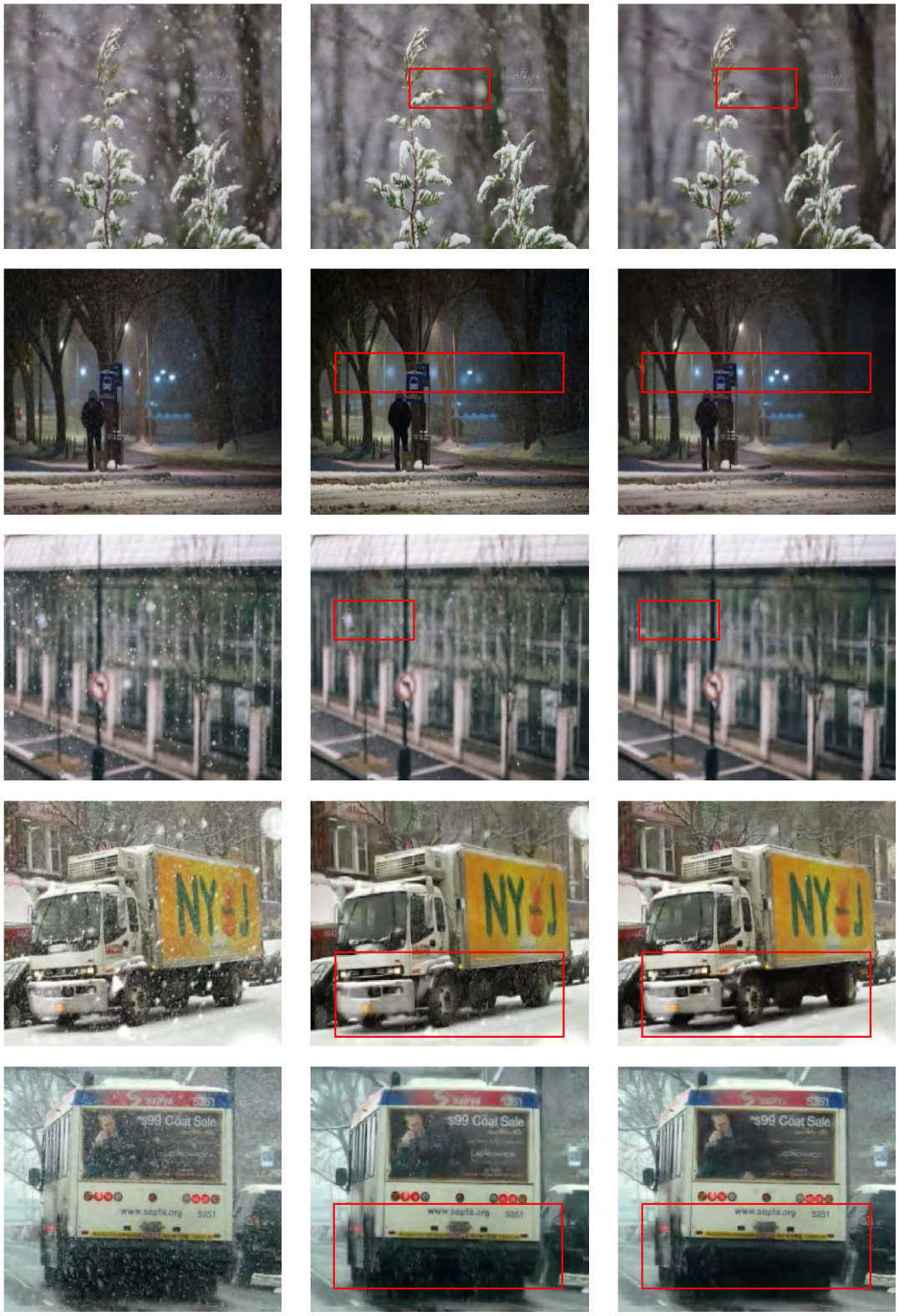

FIGURE 5.2. Real world snowy images and results from state-of-the-art methods. From left to right: Input real world snowy images, results of DesnowNet, results of our method.

CHAPTER-6
CONCLUSION

In this work, we introduce a novel deep learning snow removal algorithm based on GAN. Our approach was fully different from existing snow removal methods which focus on estimating the snow mask and aberration map from the input snowy image and then use conventional method to recover clear image. To generate better results, we have proposed a multistage network including clean background module and snow mask estimate module, therefore, it can capture more useful information. We further modify the basic GAN formulation by advancing new loss function to generate clear images. The proposed method performs favourably against several state-of-the-art methods on both synthetic dataset and real-world snowy images.

CHAPTER-7
REFERENCE

[1]. L.-W. Kang, C.-W. Lin, and Y.-H. Fu, ‘‘Automatic single-image-based rain streaks removal via image decomposition,’’ IEEE Trans. Image Process., vol. 21, no. 4, pp. 1742–1755, Apr. 2012.

[2]. Y.-L. Chen and C.-T. Hsu, ‘‘A generalized low-rank appearance model for spatio-temporally correlated rain streaks,’’ in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2013, pp. 1968–1975.

[3]. Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, ‘‘Rain streak removal using layer priors,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2736–2744.

[4]. X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, ‘‘Clearing the skies: A deep network architecture for single-image rain removal,’’ IEEE Trans. Image Process., vol. 26, no. 6, pp. 2944–2956, Jun. 2017.

[5]. W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan ‘‘Deep joint rain detection and removal from a single image,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2017, pp. 1357–1366.

[6]. X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, ‘‘Removing rain from single images via a deep detail network,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 1715–1723.

[7]. Y.-F. Liu, D.-W. Jaw, S.-C. Huang, and J.-N. Hwang, ‘‘DesnowNet: Context-aware deep network for snow removal,’’ IEEE Trans. Image Process., vol. 27, no. 6, pp. 3064–3073, Jun. 2018.

[8]. J. Long, E. Shelhamer, and T. Darrell, ‘‘Fully convolutional networks for semantic segmentation,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 3431–3440.

[9]. O. Ronneberger, P. Fischer, and T. Brox, ‘‘U-Net: Convolutional networks for biomedical image segmentation,’’ in Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. Cham, Switzerland: Springer, 2015, pp. 234–241.

[10]. K. He, G. Gkioxari, P. Dollár, and R. Girshick, ‘‘Mask R-CNN,’’ in Proc. IEEE Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2980–2988.

[11]. Goodfellow et al., ‘‘Generative adversarial nets,’’ in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2014, pp. 2672–2680.

[12]. M. Arjovsky, S. Chintala, and L. Bottou. (2017). ‘‘Wasserstein GAN.’’ [Online]. Available: https://arxiv.org/abs/1701.07875

[13]. X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, ‘‘Least squares generative adversarial networks,’’ in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2813–2821.

[14]. H. Chang, J. Lu, F. Yu, and A. Finkelstein, ‘‘PairedcycleGAN: Asymmetric style transfer for applying and removing makeup,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 40–48.

[15]. Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, ‘‘StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 8789–8797.

[16]. Y.-S. Chen, Y.-C. Wang, M.-H. Kao, and Y.-Y. Chuang, ‘‘Deep photo enhancer: Unpaired learning for image enhancement from photographs with GANs,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) 2018, pp. 6306–6314.

[17]. Ledig et al., ‘‘Photo-realistic single image super-resolution using a generative adversarial network,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 105–114.

[18]. Bulat and G. Tzimiropoulos, ‘‘Super-FAN: Integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with GANs,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 109–117.

[19]. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, ‘‘Social GAN: Socially acceptable trajectories with generative adversarial networks,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 2255–2264.

[20]. W. Deng, L. Zheng, Q. Ye, G. Kang, Y. Yang, and J. Jiao, ‘‘Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 994–1003.

[21]. K. He, X. Zhang, S. Ren, and J. Sun, ‘‘Deep residual learning for image recognition,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[22]. Ulyanov, A. Vedaldi, and V. Lempitsky. (2016). ‘‘Instance normalization: The missing ingredient for fast stylization.’’ [Online]. Available: https://arxiv.org/abs/1607.08022

[23]. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘‘ImageNet classification with deep convolutional neural networks,’’ in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2012, pp. 1097–1105.

[24]. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, ‘‘Dropout: A simple way to prevent neural networks from overfitting,’’ J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, 2014.

[25]. [Online]. Available: https://www.flickr.com/services/api/

[26]. [Online]. Available: https://tensorflow.google.cn/

[27]. J. Redmon and A. Farhadi. (2018). ‘‘YOLOV3: An incremental improvement.’’ [Online]. Available: https://arxiv.org/abs/1804.02767