Image Rain Removal Using Conditional Generative Networks Incorporating

Fangyan Zhang, Xinzheng Xu, Peng Wang

School of Intelligent Engineering and Technology, Ningxia University, Yinchuan, China
Email: lucia@nxu.edu.cn, xuxinzh@163.com, wangpstar@nxu.edu.cn

Abstract

The research of removing rain from pictures or videos has always been an important topic in the field of computer vision and image processing. Most noise reduction methods more or less remove texture details in rain-free areas, resulting in an over-smoothing effect in the restored background. The research on image noise removal is very meaningful. We exploit the powerful generative power of a modified generative adversarial network (CGAN) by enforcing an additional condition that makes the derained image indistinguishable from its corresponding ground-truth clean image. An efficient and lightweight attention machine mechanism NAM is introduced in the generator, and an IDN-CGAN model is proposed to capture image salient features through attention operations. Taking advantage of the mutual information in different dimensions of the features to further suppress insignificant channels or pixels to ensure better visual quality, we also introduce a new fine-grained loss function in the generator-discriminator pair, predicting and real data degree of disparity to achieve improved results.

Keywords

1. Introduction

In this era of ubiquitous use of mobile phones, images captured by mobile phone cameras in adverse weather conditions will degrade, greatly affecting the visual quality of the images. To improve the overall quality of these degraded images and ensure the performance of enhanced vision algorithms. Commonly used computer vision algorithms, such as autonomous driving [1], semantic segmentation [2], and object tracking [3], all require clean images as input and thus

tend to fail in bad weather conditions. Image deraining and noise removal to restore clean images are indispensable preprocessing processes for these computer vision algorithms. Specifically, image deraining and noise removal focus on removing rain streaks [4] and denoising [5] from images, respectively, by solving a linear decomposition problem. Mathematically, a rainy image can be decomposed into two separate images: one corresponding to the rain streaks and the other corresponding to the clean background image (see **Figure 1**).

Recent studies on image noise removal are based on data-driven methods. Whereas data-driven methods, such as [6], enable models to learn more robustly and flexibly with the help of large labeled datasets, which often results in robust models with better performance. But it is not very friendly to the details of rain removal. In this paper, we study the effectiveness of conditional generative adversarial network (GAN) in solving this problem. Specifically, we propose an integrated attention mechanism NAM's De-raining NAM Conditional Generative Adversarial Network (IDN-CGAN) uses a conditional GAN framework to visually enhance images degraded by rainfall.

2. Related Theories

2.1. Generative Adversarial Networks: Generative Adversarial Nets

Generative Generative Adversarial Nets [7] (Generative Adversarial Nets, GAN) is the product of the combination of game theory and deep learning. It is an unsupervised probability distribution learning method that learns the distribution of real data, generating new datasets with high similarity. GAN consists of generator and discriminator. The generator learns the distribution of real sample data and generates the most realistic fake data. The discriminator is essentially a dual classifier that needs to identify whether the input data is a real sample or fake data generated by the generator. The main purpose of the generator is to try to generate fake data that is similar to the real data, and the main purpose of the discriminator is to try to distinguish the real data from the fake data. Therefore, the training process of Generative Adversarial Network is like a game game. In all possible function sums, find the equilibrium solution of both sides.

In order to learn the data distribution of the generator on the data x, first define a random noise variable z, which can be mapped to the corresponding data

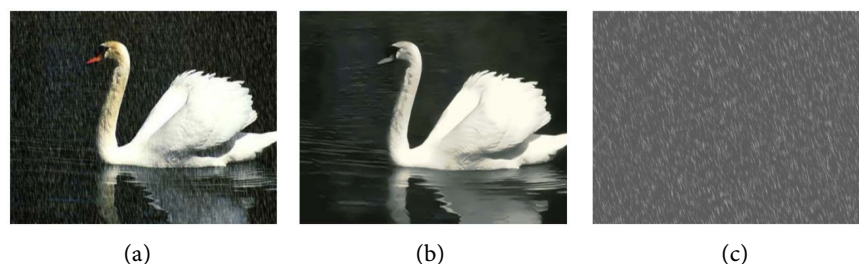|		
(a)	(b)	(c)

Figure 1. Removing rain streaks in an image. The rain image (a) can be seen as a superposition of the clean background image (b) and the rain streak image.

space by the generator, which is defined as the function of the multi-layer perceptron representation in the generative adversarial model. Because it is the function represented by the second multilayer perceptron. Where D(x) represents the real data x. When training the model, we use Dl to take the maximum value of the probability value of the real sample and the generated sample, so it can be minimized to the training summation. In this case, the value function is played by the discriminator and the generator. The optimal solution has been reached, and an equilibrium solution for both sides is found.

$$\min_{G} \max_{D} V(D,G) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (1)$$

2.2. Conditional Generative Adversarial Networks

The CGAN is the abbreviation of Conditional Generative Adversarial Nets [8], also known as Conditional Generative Adversarial Network. CGAN is an extended model of the original GAN, which adds a condition to the Generative Adversarial Network, making the Generative Adversarial Network a supervised learning. Both the generator and the discriminator add the corresponding information y as the training condition. In the generative model, the prior input noise $p(z)$ and the conditional information y together form the joint hidden layer representation. In this way, in the Formula (2) mentioned above, a condition is added to allow the network to perform a minimax game under the condition y:

$$\min_{G} \max_{D} V(D,G) = E_{x \sim p_{data}(x)}[\log D(x/y)] + E_{z \sim p_z(z)}[\log(1 - D(G(x/y)))] \quad (2)$$

2.3. Attention Mechanism NAM

Attention mechanism is one of the hotspots of research in recent years. It helps deep neural networks suppress less significant pixels or channels. The Attention Mechanism (Normalization-based Attention Module, NAM) [9] proposed in 2021 is to use the contribution factor of the weight to improve the attention mechanism. Using a batch-normalized scale factor, utilizes a variance measure of the weights of the trained model to highlight salient features. This avoids adding the fully connected and convolutional layers used in (Bottlenet Attention Module, BAM) and (Convolutional Block Attention Module, CBAM).

NAM as an efficient and lightweight attention mechanism. The module integration from CBAM was adopted and the channel and spatial attention submodules were redesigned. A NAM module is embedded at the end of each network block. For residual networks, it is embedded at the end of the residual structure. For the channel attention submodule, a scale factor from batch normalization (BN) is used, as shown in publicity (3). The scale factor measures the variance of the channels and indicates their importance.

$$B_{out} = BN(B_{in}) = \gamma \frac{B_{in} - \mu_{B}}{\sqrt{\sigma_{B}^{2} - \varepsilon}} + \beta \quad (3)$$

Among them, μ_{B} and σ_{B} are the mean and standard deviation of the mini-

mum batch B, respectively γ and β are trainable affine transformation parameters (scale and displacement). The channel attention sub-module is shown in **Figure 2** and Equation (4), where M_c represents the output feature. γ is the scale factor of each channel, and the weight is $W_\gamma = \gamma_i / \sum_{j=0} \gamma_j$. A scale factor of BN is applied to the spatial dimension to measure the importance of pixels. Name it Pixel Normalization. The corresponding spatial attention sub-module is shown in **Figure 3** and Equation (5), where the output is represented as Ms, λ is the scale factor, and the weight is $W_\lambda = \lambda_i / \sum_{j=0} \lambda_j$.

To suppress less significant weights, we add a regularization term to the loss function, as shown in Equation (6). where *x* is the input, *y* is the output, and *W* is the network weight, $l(\cdot)$ represents the loss function, $g(\cdot)$ is the l_1-norm penalty function, and *p* is the penalty function of the balance functions $g(\gamma)$ and $g(\lambda)$.

$$M_c = sigmoid(W_\gamma(BN(F_1))) \tag{4}$$

$$M_s = sigmoid(W_\lambda(BN_s(F_2))) \tag{5}$$

$$loss = \sum_{(x,y)} l(f(x,W),y) + p\sum g(\gamma) + p\sum g(\lambda) \tag{6}$$

3. The Proposed Method

3.1. Defining the Loss Function

Generative Adversarial Networks are not stable during training, we address this problem by introducing a perceptual loss into the network. A new refinement
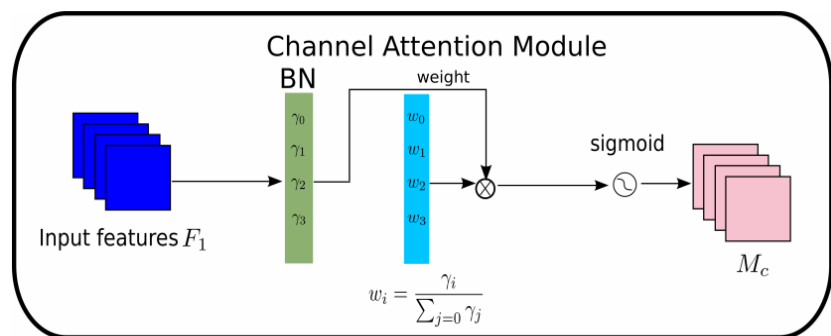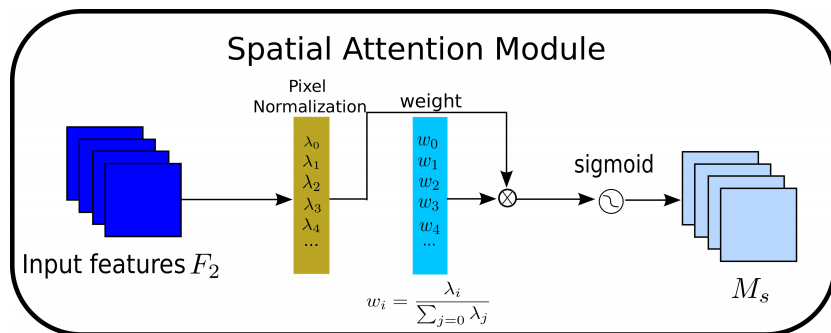

Figure 2. Channel attention mechanism.

Figure 3. Spatial attention mechanism.

loss function is proposed. Specifically, we combine pixel-to-pixel loss, perceptual loss, and adversarial loss with appropriate weights to form our new refinement loss function. The new loss function is then defined as follows:

$$L_{RP} = \lambda_e L_E + \lambda_a L_A + \lambda_p L_P \tag{7}$$

where L_A represents the adversarial loss (loss of the discriminator D) function, L_p is the perceptual loss function, L_E is the normal per-pixel loss function. Here, λ_e, λ_a, and λ_p are the predefined weights for perceptual loss and adversarial loss, respectively. If we set both λ_a and λ_p to 0, then the network reduces to a normal CNN configuration. If λ_p is set to 0, the network will reduce to a normal GAN. If λ_a is set to 0, the network reduces to the structure proposed in [10]. The three loss functions are defined as follows. Given an image $\{x, y_b\}$ with C channel. Given an image $\{x, y_b\}$ with C channel, width W and height H (*i.e.* $C \times W \times H$), where is the input image x and y_b is the corresponding ground truth, the per-pixel loss function is defined as:

$$L_E = \frac{1}{CWH} \sum_{c=1}^{C} \sum_{x=1}^{W} \sum_{y=1}^{H} \left\| \Phi E(X)^{c,w,h} - (y_b)^{c,w,h} \right\|_2^2 \tag{8}$$

where ΦE is the learning network G used to generate the derained image output. Suppose the output size of some advanced layers is $C_i \times W_i \times H_i$. Likewise, the perceptual loss is also defined as:

$$L_P = \frac{1}{C_i W_i H_i} \sum_{c=1}^{C_i} \sum_{w=1}^{W_i} \sum_{h=1}^{H_i} \left\| V(\Phi E(X))^{c,w,h} - V(y_b)^{c,w,h} \right\|_2^2 \tag{9}$$

Among them, V represents a nonlinear CNN transformation. Our goal is to minimize the distance between features. Given a set of N rain images.

3.2. Generators with Symmetrical Structure

Since the goal of single image rain removal is to generate pixel-level rain removal images, the generator should be able to remove rain streaks as much as possible without losing the detailed information of the background image. Therefore, the key to removing rain from a single image is to design a good generated by generator $\{\Phi E(X)\}_{i=1}^{N}$, the entropy loss of the discriminator is defined as:

$$L_A = -\frac{1}{N} \sum_{i=1}^{N} \log(D(\Phi E(X))) \tag{10}$$

Conditional Generative Network Incorporating Attention Mechanism NAM

In this section, we design a generative adversarial network model by incorporating the conditional generative network of the attention mechanism NAM for the problem of image rain removal. The entire network model architecture consists of two main parts, the first part generates images that are similar to the original images. The second part is to judge the most similar generated image by the contextual semantics of the original image, and iterate the first two steps through back-propagation. In this paper, the conditional generation network incorpo-

rating the attention mechanism NAM for image deraining model IDN-CGAN is shown in **Figure 4**, which clearly expresses the core idea of this chapter. The image features obtained by IDN-CGAN have good generality and generalization ability.

Since the goal of single image rain removal is to generate pixel-level rain removal images, the generator should be able to remove rain streaks as much as possible without losing the detailed information of the background image. Therefore, the key to removing rain from a single image is to design a good structure to generate a rain-removing image. Existing solutions, such as sparse coding-based methods [11] [12] [13], and neural network-based methods [14], all adopt a symmetric (encoding-decoding) structure. In these methods, a symmetric structure is used to form the generator network. The generator G directly learns the end-to-end mapping from the input rain image to the corresponding ground truth. In contrast to existing adversarial networks that use U-Net [15] [16] or ResNet [17] for occlusion-image-to-image translation in the generator, we use the recently introduced densely connected block [18]. These dense blocks allow gradients to flow, increasing parameter efficiency. Each layer in the dense block consists of three consecutive operations, batch normalization (BN), leaky rectified linear unit with leaky linear rectification function (Leaky ReLU), and 3×3 convolution. Each dense block is followed by a transition block (T) that functions as an upsampling (Tu), downsampling (Td), or no-sampling operation (Tn).

Furthermore, we embed a NAM module at the end of the generator network to effectively utilize different levels of features and guarantee better convergence.

3.3. Multi-Scale Discriminator

This paper proposes a new multi-scale discriminator. This is inspired by the use of multi-scale features in objection detection [19] and semantic segmentation [20]. Similar to the structure proposed in [15], a convolutional layer with batch normalization and PReLU activation is used as the basis throughout the
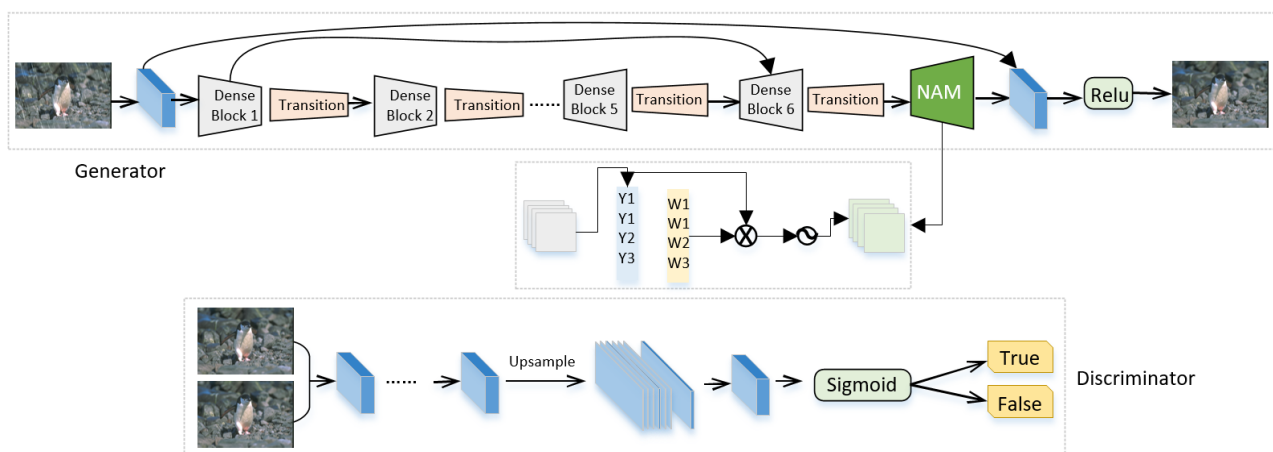

Figure 4. IDN-CGAN network frame diagram.

discriminator network. Then, multi-scale pooling modules with different scale features are stacked at the end of the discriminator. The merged features are then upsampled and concatenated, followed by a 1×1 convolution and a sigmoid function to produce a normalized probability score between [0, 1]. By using features at different scales, we explicitly incorporate global hierarchical context into the discriminator. A scaling operation weights the normalized weights to the features of each channel, and prevents the adversarial network from collapsing and oscillating during training, improving stability.

4. Experiments and Results

In this section, we detail the experiments and experimental results used to evaluate the proposed IDN-CGAN method. And the proposed method is compared with the recent state-of-the-art methods.

4.1. Dataset

The Due to the lack of large datasets for single-image rainfall training and evaluation, in order to verify the effectiveness of the proposed algorithm, we use not only real datasets but also synthetic datasets in our experiments. The training set consists of 800 images in total, divided into two types: severe rain images and light rain images, to ensure rain pixels with different intensities and directions, and generate different training and test sets. To demonstrate the effectiveness of the method on real-world data, we download a dataset of 50 rain images from the Internet. When creating this dataset, we took every care possible to ensure that the collected images differ in content, intensity and orientation of rain pixels. A more comprehensive effect has been achieved.

4.2. Evaluation Method

We employ Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [21], and Visual Information Fidelity (VIF) [22] to evaluate the performance of different methods. All of these quantitative measurements are calculated using the luminance channel. These three criteria are often used to verify the performance of the network model and the robustness of the model. The peak signal to noise ratio (PSNR) of an image is given by the following formula:

$$PSNR = 10 \log_{10} \left(\frac{MAX^2}{MSE} \right) \tag{11}$$

SSIM is used to judge whether the image is distorted, the objective standard of image noise level, and the maximum unit is expressed in decibels. The structural similarity index [21] (Structural similarity index, SSIM) is shown in the following formula:

$$SSIM = \frac{(2\mu_x \mu_y + C_1)(2\sigma_x \sigma_y + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{12}$$

where *x, y* are the uncompressed undistorted image and the image contrasted by

x, $\mu_x\mu_y$ is a function of brightness, $\sigma_x\sigma_y$ is a function of contrast, the constant C_1 is to ensure the stability of the system at $\mu_x^2 + \mu_y^2 \Rightarrow 0$, and C_2 is also a constant $C_3 = C_2/2$. In the evaluation index, the larger the value of PSNR, VIF and SSIM, the better the image restoration and the smaller the image distortion.

4.3. Experimental Results

The experimental parameter settings in this paper are done on the Tensorflow learning platform. The experiment was performed on a desktop computer, Intel-i5 3.3G CPU, 8GB RAM, Windows 10, TensorFlow 1.1, Opencv2.0, the learning rate was set to 0.5, the number of iterations was 50, and the batch size was 2. This paper selectively samples heavy rain images to show that our method performs well under difficult conditions. The performance of the proposed method and recent state-of-the-art methods is evaluated on real-world rainfall test images. The deraining results of all methods on the derained images input by two samples are shown in **Figure 5**. By comparing the proposed method with other methods, by observation, we can clearly observe that GMM [23] tends to add artifacts on derained images. Although CCR [24] is able to remove rain streaks, it produces blurry results, which are not visually appealing. JORDER [25] can reduce the intensity of rain or remove streaks in some areas, however, they cannot completely remove rain streaks. Raindrops are still visible in the magnified region of interest despite other methods, and good visual performance can be achieved. Compared with other methods, this method is able to successfully remove most of the rain streaks while maintaining the details of the derained images.

The three evaluation parameters PSNR, SSIM, UQI and IDN-CGAN are all learned by using training images of synthetic training datasets. Using the test images of the synthetic datasets discussed earlier, as shown in **Table 1**, it can be

Figure 5. Comparison of experimental results.

Table 1. Comparison with other methods using three different evaluation criteria for heavy rain images and light rain images.

heavy rain images	GMM [23]	JORDER [25]	CCR [24]	**IDN-CGAN**
PSNR	20.21	22.98	21.45	**23.76**
SSIM	0.7657	0.8012	0.7964	**0.8317**
VIF	0.2821	0.3327	0.3873	**0.39582**
light rain image	GMM [23]	JORDER [25]	CCR [24]	**IDN-CGAN**
PSNR	20.97	20.83	22.26	**24.67**
SSIM	0.7865	0.8263	0.8077	**0.8256**
VIF	0.3654	0.3264	0.4002	**0.4158**

seen that the introduction of adversarial loss improves the visual quality compared to the traditional CNN architecture. For comparison: the first indicator is the peak signal-to-noise ratio (PSNR). The PSNR of the model IDN-CGAN proposed in this paper is significantly higher than the other three methods in the heavily derained image and the lightly derained image. The second indicator is The estimated overall structural similarity index (SSIM), the SSIM values of GMM [23] and CCR [24] are both lower than JORDER [25], while the SSIM value of IDN-CGAN is slightly higher than that of JORDER [25]; Authenticity (VIF), although IDN-CGAN performs better than the other three models, the difference is not large, the reason may be that the proposed method misses some rain patterns in the output image. (The higher the PSNR and VIF evaluation indicators and the higher the SSIM indicator, the clearer the model repair results and the better the effect.)

5. Summary and Outlook

To achieve clearer image deraining, we propose a conditional generative network (IDN-CGAN) incorporating an attention mechanism NAM for image deraining. In IDN-CGAN, we first use the generator fused with the attention mechanism NAM to generate sharper images, and then propose a refinement loss function that enables the discriminator model to discriminate between real images and derained images. To validate the proposed IDN-CGAN, experiments on synthetic and real datasets demonstrate the superiority and effectiveness of IDN-CGAN with limited training data. It is worth noting that there is room for further research on IDN-CGAN. One direction is to use IDN-CGAN in real-world scenarios where we can only collect a few examples. Another direction is to construct a learning task of n-frequency k-shots by dynamically clustering image patches from data batches instead of pre-clustering all image patches.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Andreas, G., Philip, L. and Raquel, U. (2012) Are We Ready for Autonomous Driving? The Kitti Vision Benchmark Suite. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3354-3361.

[2] Sachin, M., Mohammad, R., Anat, C., Linda, S. and Hannaneh, H. (2018) Espnet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation. *Proceedings of the European Conference on Computer Vision*, 552-568.

[3] Dorin, C., Visvanathan, R. and Peter, M. (2003) Kernel-Based Object Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**, 564-577. https://doi.org/10.1109/TPAMI.2003.1195991

[4] Fu, Y.-H., Kang, L.-W., Lin, C.-W. and Hsu, C.-T. (2011) Single-Frame-Based Rain Removal via Image Decomposition. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1453-1456. https://doi.org/10.1109/ICASSP.2011.5946766

[5] Kai, Z., Zuo, W. and Zhang, L. (2018) Ffdnet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising. *IEEE Transactions on Image Processing*, **27**, 4608-4622. https://doi.org/10.1109/TIP.2018.2839891

[6] Fu, X., Huang, J., Zeng, D., Huang, Y., Ding, X. and John, P. (2017) Removing Rain from Single Images via a Deep Detail Network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3855-3863. https://doi.org/10.1109/CVPR.2017.186

[7] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., *et al.* (2014) Generative Adversarial Networks: Advances in Neural Information Processing Systems. **3**, 2672-2680.

[8] Radford, A., Metz, L. and Chintala, S. (2015) Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks.

[9] Liu, Y.C., Shao, Z.R. and Teng, Y.Y. (2021) NAM: Normalization-Based Attention Module.

[10] Johnson, J., Alahi, A. and Li, F.F. (2016) Perceptual Losses for Realtime Style Transfer and Super-Resolution. *European Conference on Computer Vision*, Springer, 694-711. https://doi.org/10.1007/978-3-319-46475-6_43

[11] Starck, J.-L., Elad, M. and Donoho, D.L. (2005) Image Decomposition via the Combination of Sparse Representations and a Variational Approach. *IEEE TIP*, **14**, 1570-1582. https://doi.org/10.1109/TIP.2005.852206

[12] Kang, L.-W., Lin, C.-W. and Fu, Y.-H. (2012) Automatic Single-Image-Based Rain Streaks Removal via Image Decomposition. *IEEE TIP*, **21**, 1742-1755. https://doi.org/10.1109/TIP.2011.2179057

[13] Bobin, J., Starck, J.L., Fadili, J.M., Moudden, Y. and Donoho, D.L. (2007) Morphological Component Analysis: An Adaptive Thresholding Strategy. *IEEE Transactions on Image Processing*, **16**, 2675-2681. https://doi.org/10.1109/TIP.2007.907073

[14] Xie, J., Xu, L. and Chen, E. (2012) Image Denoising and Inpainting with Deep Neural Networks. *NIPS*, 341-349.

[15] Isola, P., Zhu, J.-Y., Zhou, T. and Efros, A.A. (2017) Image-to-Image Translation with Conditional Adversarial Networks. *CVPR*. https://doi.org/10.1109/CVPR.2017.632

[16] Ronneberger, O., Fischer, P. and Brox, T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 234-241. https://doi.org/10.1007/978-3-319-24574-4_28

[17] Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., *et al.* (2017) Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1-8. https://doi.org/10.1109/CVPR.2017.19

[18] Huang, G., Liu, Z., van der Maaten, L. and Weinberger, K.Q. (2017) Densely Connected Convolutional Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2017.243

[19] He, K., Zhang, X., Ren, S. and Sun, J. (2014) Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. European Conference on Computer Vision. Springer, 346-361. https://doi.org/10.1007/978-3-319-10578-9_23

[20] Zhao, H., Shi, J., Qi, X., Wang, X. and Jia, J. (2017) Pyramid Scene Parsing Network. *Proceedings of the IEEE International Conference on Computer Vision*, 1-8. https://doi.org/10.1109/CVPR.2017.660

[21] Wang, Z., Bovik, A.C., Sheikh, H.R. and Simoncelli, E.P. (2004) Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE TIP*, **13**, 600-612. https://doi.org/10.1109/TIP.2003.819861

[22] Sheikh, H.R. and Bovik, A.C. (2006) Image Information and Visual Quality. *IEEE TIP*, **15**, 430-444. https://doi.org/10.1109/TIP.2005.859378

[23] Li, Y., Tan, R.T., Guo, X., Lu, J. and Brown, M.S. (2016) Rain Streak Removal Using Layer Priors. 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), June 2016, 2736-2744. https://doi.org/10.1109/CVPR.2016.299

[24] Zhang, H. and Patel, V.M. (2017) Convolutional Sparse and Low-Rank Coding-based Rain Streak Removal. 2017 *IEEE WACV*, IEEE, 1-9. https://doi.org/10.1109/WACV.2017.145

[25] Yang, W., Tan, R.T., Feng, J., Liu, J., Guo, Z. and Yan, S. (2017) Deep Joint Rain Detection and Removal from a Single Image. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1357-1366. https://doi.org/10.1109/CVPR.2017.183