Research Article

Research of Single Image Rain Removal Algorithm Based on LBP-CGAN Rain Generation Method

Ping Xue1,2 and Hai He2

1The Higher Educational Key Laboratory for Measuring & Control Technology and Instrumentation of Heilongjiang Province, Harbin University of Science and Technology, Harbin 150080, China
2School of Automation, Harbin University of Science and Technology, Harbin 150080, China

Correspondence should be addressed to Ping Xue; xueping@hrbust.edu.cn

Received 26 September 2020; Revised 16 June 2021; Accepted 6 July 2021; Published 14 July 2021

Academic Editor: Akemi Galvez

Copyright © 2021 Ping Xue and Hai He. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Rain has an undesirable negative effect on the clarity of the collected images. In situations where images are captured in rain, it can lead to a loss of information and disability in reflecting real images of the situation. Consequently, rain has become an obstacle in outdoor scientific research studies. The reason why images captured in rain are difficult to process is due to the indistinguishable interactions between the background features and rain textures. Since current image data are only processed with the CNN (convolutional neural network) model, a trained neural network to remove rain and obtain clear images, the resulted images are either insufficient or excessive from standard results. In order to achieve more ideal results of clearer images, series of additional methods are taken place. Firstly, the LBP (local binary pattern) method is used to extract the texture features of rain in the image. Then, the CGAN (conditional generative adversarial network) model is constructed to generate rain datasets according to the extracted rain characteristics. Finally, the existing clear images, rain datasets generated by CGAN, as well as the images with rain are used for convolution operation to remove rain from the images, and the average value of PSNR (peak signal to noise ratio) can reach 38.79 by using this algorithm. Moreover, a large number of experiments are done and have proven that this joint processing method is able to successfully and effectively generate clear images despite the rain.

1. Introduction

With the continuous development of computer vision systems, more convenient methods are provided for outdoor exploration and target tracking. However, the outdoor weather is complex and changeable, especially the severe rain and snow weather. It seriously affects the image quality collected by the vision system, which can mislead the outdoor workers with inaccurate information [1, 2]. Therefore, it is necessary to solve the problem by removing rain from rainy images [3]. However, the rainy image background scene varies, and the density, shape, and direction of the rain are also diverse. So, the problem of removing rain from images has become an important challenge in the field of computer vision.

Presently, image denoising and rain removal have always been classic problems in the field of computer science. Rain images can also be regarded as a special image noise. The earliest research of the rain removal algorithm is to remove the rain in the video [4]. Due to the temporal and spatial relativity of the video sequence, image characteristics can be extracted from the video sequence of the same pixel at consecutive times to determine the position of the pixel affected by rain. Therefore, it is relatively simple to remove the rain from a video. Starik and Werman [5] proposed the method of time-domain median filtering to remove rain from video images, which led to the development of research on image removal from rain. As for the rain removal research of a single image, because a single image does not have a video sequence that can be used, rich information brought by the
In this article, aiming at the lack of consideration of the relationship between rain features and background in current rain removal methods [18–20], we analyze the pixel features of the rain images, clean images, and rain streak images. Because of the obvious overlap in pixel statistics, it is difficult to separate them from pixel level by physical means. Therefore, an LBP (local binary pattern) method is proposed to extract the texture features of rain in the image [21]. According to the extracted approximate rain features and combined with the background image, a CGAN (conditional generative adversarial network) is designed to automatically learn the relationship between rain features and the background to generate the rain datasets. Finally, the convolution operation is performed according to the rain image, the clean image, and the generated rain dataset to realize the rain removal of the rain images.

2. Analysis of Rain Removal and Design Principle of Rain Removal Algorithm

2.1. Single Image Rain Feature Description. In the research of image rain removal, the rain image (marked as O) is usually defined as the linear superposition of the rainless background image (marked as B) and the rain streak image (marked as R), which is O = B + R [22, 23]. According to this principle, statistical analysis is performed on the pixel features of the background component and the rain component in the rain image, as shown in Figure 1. From the statistical results of pixel distribution, it can be seen that each image has an obvious overlap in pixels, which indicates that there is a mutual influence between rain and background. Therefore, it is difficult to separate rain and background by using frequency-domain transform for feature comparison.

In the actual rain removal research of the rain images, due to the diversity of the background, a single deep learning network is used to generate the rain-removed images based on the input rain images. Due to the lack of consideration of the interaction between the background and the rain, it is easy to cause sparse rain lines or image distortion in the process of going to rain. The background features are difficult to grasp due to the diversity of image backgrounds. Relatively speaking, the features of rain are mainly reflected by the shape, size, and direction. It is easier to grasp the characteristics of rain.

2.2. Design Principle of Rain Removal Algorithm. Based on the analysis of the rain removal problem in the rain image, considering the mutual influence between the rain and the background in rain images and the diversity of background features in different rain images, it is difficult to grasp the background features. However, the features of rain mainly include morphological information such as shape, direction, and size, as well as sensory information such as sparse and dense rain lines, which are relatively easy to grasp in different background images. So, the rain removal model is built, as shown in Figure 2.

First of all, the LBP method is introduced to extract the texture features of the rain from the rain images in this
article. When the LBP method is used directly, the extracted rainwater texture features are susceptible to interference from other texture features in the background. Therefore, the circle and rotation invariance theory is introduced, and the LBP algorithm is used to improve the accuracy of rain feature extraction. And then according to the features of the shape, size, and direction of the obtained rainwater, combined with the rain image, the GAN is designed to generate rain datasets. Considering the interaction between rain and the background, it affects the rain images. On this basis, the generator is trained from the perspective of conditional confrontation loss, and a gradient penalty factor is introduced to constrain the discriminator. The CGAN is reinforced to learn the mutual influence between the rain and the background to generate the rain feature dataset (marked as D). Finally, a rain removal model is constructed to realize the rain

Figure 1: Histogram analysis of intensity distribution in color channels: (a) pixel feature statistical analysis about the background component; (b) pixel feature statistical analysis about the image with rain; (c) pixel feature statistical analysis about the rain component.

Figure 2: Joint rain removal model.
removal of the rainy images combining the background image, the rain streak image (marked as R), the rainy feature dataset (marked as D), and the rain image (marked as O).

3. Image Rain Removal Algorithm

Model Construction

3.1. Extraction Algorithm of Rain Texture Features Based on LBP. The local color distribution and light intensity distribution will be affected in rain images due to rain interference. Therefore, the extracted rain streaks feature will be fully utilized to achieve rain removal in rain images in this paper’s rain removal solution. LBP (local binary pattern), an operator used to describe the local texture features of an image, has the advantages of rotation and grayscale invariant texture classification. LBP, first proposed by Ojala et al. [24, 25], was used for texture feature extraction. Later, it was combined with the HOG feature classifier to improve the detection effect on some datasets [26, 27]. LBP is used in this paper to extract rain features from rain images. The LBP algorithm combined with the extracted features is constrained by the rotation-invariant method so that the extracted image features are suitable for GAN to generate a reliable rain image dataset for rain removal of rain images.

The LBP value can be used to reflect the texture information of the area, and the original LBP operator is defined in a $3 \times 3$ window. Take the central pixel of the window as the threshold and compare it with the gray values of the adjacent 8 pixels. If the surrounding pixel value is greater than the central pixel value, then the position of the pixel is marked as 1; otherwise, 0. In this way, after comparing 8 points in the $3 \times 3$ neighborhood, an 8-bit binary number can be generated (usually converted to a decimal number or LBP code, a total of 256 types), and thus the LBP value of the center pixel of the window is obtained as shown in Figure 3.

The formula of the LBP algorithm is described as follows:

$$LBP(x_c, y_c) = \sum_{i=1}^{I} 2^i S(g_i - g_c),$$

(1)

where $LBP(x_c, y_c)$ is the pixel coordinate of the center point in the neighborhood of the $3 \times 3$ image, $g_i$ is the gray value of the No. $i$ pixel in the neighborhood, $g_c$ is the gray value of the center point, and $S(g_i - g_c)$ is the symbolic function. If $g_i - g_c > 0$, then $S(g_i - g_c) = 1$. If $g_i - g_c \leq 0$, then $S(g_i - g_c) = 0$.

At present, the proposed LBP operators mainly include circular LBP, rotation-invariant LBP, and uniform mode LBP based on the principle of LBP. The abovementioned operators are used to extract the texture features of rainy images. The results are shown in Figure 4.

In order to ensure the accuracy of rain texture extraction, the LBP algorithm is to operate the original image without gray processing. As shown in Figure 4, it can be seen that the nonlinear variation of illumination causes the uneven distribution of the grayscale gradient of the image. That is to say, compared with most areas of the image when a small part of the area changes drastically if only circular LBP, rotation-invariant LBP, or uniform LBP operators are adopted, it will highlight its limitations, i.e., easy to lose local details of the image. As such, this paper makes a binary improvement to the LBP algorithm (circular neighborhood enhancement and rotation invariance) to retain the detailed information with strong distinguishing ability.

(1) Feature description is carried out after extending the neighborhood of $3 \times 3$ feature points to $5 \times 5$ image areas. The 16 pixels in the center of the feature points, namely, LBP value, are used as the 16-dimensional feature description of the feature area, as shown in Figure 5.

(2) The pixels used in the circular expansion of the neighborhood range are expanded from the original 8 to 16. The calculation formula is as follows:

$$LBP_R(x_c, y_c) = \sum_{i=1}^{I} 2^i S(g_i(x_c, y_c) - g_c),$$

$$x_i = x_c + R \cos \left(\frac{2\pi i}{I}\right),$$

$$y_i = y_c + R \sin \left(\frac{2\pi i}{I}\right),$$

(2)

where $LBP_R(x_c, y_c)$ is the coordinate of the sampling point after expansion and $g_i(x_c, y_c)$ is the gray value of the sampling point.

In this extraction algorithm of rain texture features, the symbols used in the LBP formula are shown in Table 1.

(3) Based on the extended circle neighborhood, the rotation-invariant LBP is developed. As shown in Figure 6, the values obtained from the original LBP are converted into binary codes. There will be 16 situations (including itself) when it is circularly shifted. The smallest LBP value is taken, which is rotation invariant, because the rotation on the image is equivalent to the process of the above 16 kinds of shifts, and the 16 cases all correspond to the same value; that is, they have the rotation-invariant attribute.

(4) The rain streaks feature image extracted by the rotation-invariant LBP algorithm after the circle neighborhood expansion is shown in Figure 7.

Figure 7(a) is original image of rain day. Figure 7(b) is rain water, and Figure 7(c) is the result of improved LBP which is the rotation-invariant LBP algorithm. Because of the rotation invariance of the LBP algorithm and the expansion of circle neighborhood radius, the improved LBP algorithm has strong grasp of rainwater characteristics. Compared with the results of rain texture extraction in Figures 4 and 7, the nonrain texture extracted in Figure 7 is significantly reduced, and the accuracy of rainwater texture extraction is improved. Therefore, the rain texture extracted by the improved LBP algorithm can play a better role in the later rainwater removal.
3.2. Construction of Rain Model Based on CGAN. This section mainly describes a rain generation model building based on the characteristics of rain. When the existing data samples are used to train the image rain removal network, due to the lack of consideration of the rain details, the rain image is either insufficient or overfitting to remove the rain. Therefore, to improve the performance of rain removal, modeling is carried out according to the actual rain streaks features, and then more feature data are generated to form a rain feature dataset, which can obtain samples with more diversity and arbitrariness.

Given that GAN can learn the distribution of complex features [28], the GAN is used to estimate the distribution of rain to generate a larger rain sample dataset.

The traditional GAN is mainly composed of generator $G$ and discriminator $D$. The generator $G$ is used to capture the data distribution, and the discriminator $D$ is used to estimate the probability that a sample comes from real data instead of generated data. The rain feature of rain images is easily disturbed by environmental factors, so if the traditional GAN is directly used, to be honest, the environmental conditions are not taken into full consideration. Therefore, when building a GAN, additional external information can be used as part of the input layer to build a CGAN (conditional generative adversarial network).

The CGAN architecture studied in this paper uses a fully convolutional network, U-Net, to replace the MLP fully connected network because U-Net ensures that the generated rain dataset is consistent with the underlying structure of the background image to the greatest extent.
random noise $z$ to the generated image $x$. However, according to the analysis of this paper, the interaction between the rain feature and the background image will affect the image’s rain removal effect. Therefore, this paper takes the rain texture feature $f$ and background image $b$ extracted by LBP as additional input information so as to achieve the mapping of the given random noise $z$ to the generated image $x$ under the conditions $f$ and $b$.

The designed CGAN optimization process can be summarized as a two-person minimax game with conditional probability in which $G$ and $D$ train at the same time. It is to minimize the expectation of $\log (1 - D(G(z|f,b)))$ by fixing discriminator $D$ and adjusting the parameters of $G$. And then, it is to maximize the expectation of $\log D(x|f,b) + \log (1 - D(G(z|f,b)))$ by fixing generator $G$ and adjusting the parameters of $D$. Its objective function is as follows:

$$
\min_G \max_D V(D,G) = E_{x \sim P_{data}}[\log D(x|f,b)] + E_{z \sim P_z}[\log (1 - D(G(z|f,b)))].
$$

(3)

According to the input rain features, the generator generates real enough samples so that the discriminator considers them as real samples. Based on the constructed CGAN objective function, this paper builds the conditional adversarial loss function $L_G$ of the generator and the discriminator conditional adversarial loss function $L_D$ with gradient penalty factor, respectively, in order to ensure that the generated rain sample set can reflect the influence of background on rain detail characteristics.

The traditional generator loss function of the GAN mainly considers the loss of mapping $G: Z \rightarrow x$ from the random noise $z$ to the generated image $x$. The CGAN constructed in this paper needs to ensure that the generated rain dataset can fully reflect the relationship between the background image and the rain noise. As such, this paper adopts the mapping relationship of $G: (z|f,b) \rightarrow x$ according to the input background image $b$ and the extracted rain feature $f$ and takes into account that the part of the generated image without rain is not affected by the generated rain-drops. Starting from the two aspects of conditional mapping and content loss, the following conditional adversarial loss function is constructed to enrich the details of the generated image:

$$
L_G = -E_{z \sim P_z}D(G(z|f,b)) + E_{z \sim P_z}[\|x - G(z|f,b)\|_p].
$$

(4)

Among them, $-E_{z \sim P_z}D(G(z|f,b))$ is the conditional adversarial loss and $E_{z \sim P_z}[\|x - G(z|f,b)\|_p]$ is the content loss of the generated image.

The discriminator needs to score the input real sample and the generated sample, and its goal is to give a lower evaluation to the generated sample and a higher evaluation to the real one. As for the loss function of the discriminator, a gradient penalty factor is introduced into the input of the discriminator on the basis of the condition loss of the discriminator in order to avoid the influence of the premature loss of the discriminator on the generator. As a result, the loss function of the discriminator includes the real sample loss, the generated sample loss, and the gradient

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure7.png}
\caption{The result of the improved LBP algorithms: (a) original rain image; (b) rainwater; (c) result of improved LBP.}
\end{figure}
penalty loss term. And the loss function of the discriminator can be expressed as follows:

\[ L_D = -E_{z \sim P_d}(z)D(G(z | (f, b))) - E_{x \sim P_data}(x)D(x | (f, b)) + \gamma E_{x \sim P_data}(x)\left[\|\nabla_x D(\tilde{x}, x | (f, b))\|_p - 1\right]^2. \]  

(5)

Among them, the generated sample loss is \(-E_{z \sim P_d}(z)D(G(z | (f, b)))\), the real sample loss is \(-E_{x \sim P_data}(x)D(x | (f, b))\), the gradient penalty loss term is \(\gamma E_{x \sim P_data}(x)\left[\|\nabla_x D(\tilde{x}, x | (f, b))\|_p - 1\right]^2\), and \(\gamma\) is the weight coefficient for balancing different loss functions.

The symbols used in CGAN are shown in Table 2.

In accordance with the above analysis of the CGAN conditional loss function, combined with the input and output of the generator and the discriminator and the corresponding conditional loss function, the CGAN rain dataset generation model is built under the U-Net framework as shown in Figure 8.

The generator network structure is shown in Figure 8. Some low-level information is shared between the input and output images (this shared information helps to ensure the quality of the reconstructed image) to ensure that the underlying structure of the reconstructed image and the source image remains unchanged. In terms of generative adversarial networks, we adopt the encoder-decoder structure based on the U-Net [16] network. U-Net network adopts a skip connection to splice the low-level features of the decoder in the network with the high-level features obtained from the corresponding decoding layer and then serve as the input of the next layer. This method can ensure the consistency of the generated image and the underlying structure of the background to the greatest extent. In the first and last layers of the U-net network, batch normalization (BN) is not performed, while BN processing is added to other layers. LReLU (leaky-RelU) activation function is used for downsampling, while the Tanh activation function is used for upsampling except for the last layer, and the RelU activation function is used for other convolution layers.

The discriminator as shown in Figure 8 has two types of inputs, the image output by the generator and the real image. In Figure 8, the network structure of the discriminator is used to generate samples for input. At the same time, the conditional adversarial network is adopted and the rain image is taken as the condition. Through channel splicing, the condition is spliced with the reconstructed image output by the generator and then input into the discriminator together. The discriminator network will eventually output a scalar, representing the score of the input by the discriminator network. In the discriminator network structure of Figure 1, LReLU is adopted to the activation function of the convolution layer. Except for the first layer and the last layer, BN is used in other convolution layers.

The rain data are used to train CGAN and the traditional GAN designed in this paper. The comparison of the loss curves between the generator and the discriminator in the training process is shown in Figure 9.

### Table 2: The symbols of CGAN.

| Symbol | Full name                      |
|--------|--------------------------------|
| D      | The discriminator              |
| G      | The generator                  |
| L_G    | The conditional adversarial loss function of the generator |
| L_D    | The discriminator conditional adversarial loss function with gradient penalty factor, respectively |
| x      | The generated image            |
| z      | The random noise               |
| b      | The input background image     |
| f      | The extracted rain feature     |

3.3. Joint Rain Removal Algorithm Based on CGAN and CNN Convolution. At the moment, most rain removal algorithms mainly use a large number of rain images and clear image datasets to train the CNN for rain removal in the rain images, which have achieved remarkable results [29]. The CNN can learn potential rain images from the paired datasets to remove rain, but it neglects to consider the rain features in the rain images. As such, this paper adjusts the dataset input to the CNN. Eventually, combined with the principle of CNN, the image rain removal is realized in accordance with the reconstructed rain dataset, rain images, and clear images.

The general model of rain image is as follows: \(O = B + R\). \(O\) is the collected rain image, \(B\) is the background image, and \(R\) is the rain streak. Given the traditional method, the density and shape of rain in \(R\) are not fully considered. At the same time, there is no local rain removal, resulting in excessive image smoothing. Therefore, based on the rain feature dataset \(D\) and the rain streak \(R\) reconstructed by CGAN, this paper performs convolution operation and proposes a new model.

\[ O = \alpha B + (1 - \alpha) \sum_{j=1}^{J} D_j \circ R. \]  

(6)

Each \(D_j\) is the feature parameter of the No. \(j\) group of rain extracted by the generative adversarial network, and \(\alpha\) is the harmonic parameter to simulate the mutual influence between the background and the rain. The value of \(\alpha\) is adjusted according to the feedback of the operation result during the convolution operation, and the initialization value is 0.5. According to formula (6), the MAP (maximum a posteriori) estimation [30] is established for processing:

\[ \arg \min_{B, R} \|O - B - D \circ R\|_2^2 + K_B(B) + K_D(D) + K_r(R). \]  

(7)

\(K_B(B), K_D(D),\) and \(K_r(R)\) in formula (7) represent the prior knowledge of \(B, S,\) and \(R,\) respectively, which are learned from the training data during the network training process and are implicitly embedded in the network. Therefore, the joint rain removal model of CNN convolution operation in this paper is shown in Figure 10.

As shown in Figure 10, firstly, the convolution operation is performed by combining \(R\) and \(D,\) and the
Figure 8: Schematic diagram of the CGAN.

Figure 9: The comparison of the loss curves in the training process.
obtained result is convoluted with $B$. Secondly, the result of the second convolution is combined with $O$ for the next convolution. Finally, the final clear image is generated by combining the processing results of each convolution layer.

4. Analysis of Results

4.1. Experimental Environment. This experiment is based on Windows 10 operating system. The CGAN model is built under the framework of tensorflow platform of keras2.1.4. Tensorflow framework is a deep learning framework based on data flow graph computing developed by Google team in recent years. It has the advantages of high flexibility, strong portability, automatic differential solution, support for multiple programming languages, support for optimal performance, and so on. The experimental hardware parameters are shown in Table 3.

4.2. Image Rain Removal Algorithm Comparison Experiment Verification. In order to verify the rain removal effect of the algorithm used in this paper compared with other algorithms, the algorithm in this paper is compared with the rain removal algorithm in literature [8], literature [10], and literature [11], respectively, such as DID-MDN, CNN, and Pix2Pix. In the experiment, the rain vehicle image dataset, the outdoor rain landscape image dataset, and the building rain image dataset (500 for each dataset) were used to analyze the different rain removal algorithms. The result of rain removal is shown in Figure 11.

Figure 11(a) shows the original rainy images. Figure 11(b) shows the effect diagram of the DID-MDN rain removal algorithm. Figure 11(c) shows the effect of the CNN image rain removal algorithm. Figure 11(d) shows the effect diagram of the Pix2Pix graph rain removal algorithm. Figure 11(e) shows the effect diagram of the algorithm used in this paper. The results can be seen from Figure 11. The algorithm in Figure 11(b) does not remove rain well, and there are still a lot of residues. The algorithm in Figure 11(c)

4.3. Image Quality Evaluation Based on PSNR to Remove Rain. In order to evaluate the image quality of the image after the above four algorithms are processed to remove the rain, the PSNR image quality evaluation standard is used to evaluate the image quality of the reconstructed image obtained by the four algorithms [31]. Table 4 shows the PSNR value of the images generated after the rain vehicle image dataset, the outdoor rain landscape image dataset, and the building rain image dataset under different rain removal algorithms. The algorithm in this paper is compared with the rain removal algorithm in literature [8], literature [10], and literature [11], respectively, such as DID-MDN, CNN, and Pix2Pix. It can be seen that the evaluation index value of the algorithm in this paper is obviously better than the results of other algorithms for the three images after the rain, and it can restore the image visually. It can be seen from Table 4 that the average value of PSNR can reach 38.79 by using the algorithm of this paper.

\[
B_{\text{clear}} = \frac{1}{\alpha}[O - (1 - \alpha)\sum D_j o R]
\]

Figure 10: The process of convolution operation.

Table 3: Experimental hardware parameters.

| Parameter name | Parameter value          |
|----------------|--------------------------|
| CPU            | Intel Core i7 7700HQ     |
| Number of cores/threads | Four core/eight thread   |
| Running memory | 16 GB                    |
| Graphics chip  | GTX 1080 Max-Q           |
| Memory capacity| 8 GB                     |
5. Conclusions

Rain will greatly affect the clarity of the collected pictures so that the information provided by the pictures cannot reflect the situation of real things, which becomes an obstacle to the acquisition of image information for outdoor scientific research. This paper considers the interaction between rain texture and background, starting from the perspective of rain image characteristics. From the point of view of rain image features, the problem of removing rain from rainy images is studied. Firstly, the local binary pattern (LBP) method is used to extract the texture features of the rain from rain image, and the LBP algorithm is made better based on the circle and rotation invariance theory to improve the accuracy of the rain texture feature extraction. Then, a CGAN was constructed based on the extracted rainwater features. A fully convolutional U-Net network was used to replace the MLP fully connected network when constructing CGAN. This method can guarantee the generated rain dataset. This method can ensure that the generated rain dataset is consistent with the underlying structure of the background image to the greatest extent and introduces the conditional adversarial loss, content loss, and gradient penalty loss to train the network. Finally, the rain feature dataset generated by CGAN and the existing clear image as well as the image with rain is used to perform convolution operation to remove the rain in the rainy image. It can be seen that the algorithm described in this article can improve the consideration of the mutual influence between the rain texture and the background according to the experimental comparison results and the PSNR image quality evaluation standard so that the image has better details, contrast, and saturation after the rain has been removed.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the Natural Science Foundation of Heilongjiang Province of China (F201310) and the Harbin Science and Technology Innovation Talents Project (2016RAQXJ037).
References

[1] S. Du, Y. Liu, M. Ye, Z. Xu, J. Li, and J. Liu, “Single image deraining via decorrelating the rain streaks and background scene in gradient domain,” Pattern Recognition, vol. 79, pp. 303–317, 2018.

[2] K. Jiang, Z. Wang, P. Yi et al., “Multi-level memory compensation network for rain removal via divide-and-conquer strategy,” IEEE Journal of Selected Topics in Signal Processing, vol. 15, no. 2, p. 2021.

[3] Q. Yang, M. Yu, Y. Xu, and S. Cen, “Single image rain removal based on deep learning and symmetry transform,” Symmetry, vol. 12, no. 2, p. 224, 2020.

[4] T. Jayaraman and G. S. Chinnusamy, “Investigation of filtering of rain streaks affected video sequences under various quantisation parameter in HEVC encoder using an enhanced V-BMD4 algorithm,” IET Image Processing, vol. 14, no. 2, pp. 337–347, 2020.

[5] S. Starik and M. Werman, “Simulation of rain in videos,” in Proceedings of the 3rd International Workshop on Texture Analysis and Synthesis, pp. 95-100, Nice, France, 2003.

[6] Y. Luo, Y. Xu, and H. Ji, “Removing rain from a single image via discriminative sparse coding,” in Proceedings of the IEEE International Conference on Computer Vision, pp. 3397–3405, Santiago, Chile, 2015.

[7] L. Zhu, C. W. Fu, D. Lischinski et al., “Joint bi-layer optimization for single-image rain streak removal,” in Proceedings of the IEEE International Conference on Computer Vision, pp. 2526–2534, Venice, Italy, 2017.

[8] H. Zhang and V. M. Patel, “Density-aware single image deraining using a multi-stream dense network,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 695–704, Salt Lake City, UT, USA, June 2018.

[9] X. Fu, J. Huang, and X. Ding, “Clearing the skies: a deep network architecture for single-image rain removal,” IEEE Transactions on Image Processing, vol. 26, no. 6, pp. 2944–2956, 2017.

[10] W. Yang, R. T. Tan, J. Feng, Z. Guo, S. Yan, and J. Liu, “Joint rain detection and removal from a single image with contextualized deep networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 6, pp. 1377–1393, 2020.

[11] P. Isola, J. Y. Zhu, and T. Zhou, “Image-to-image translation with conditional adversarial networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134, Honolulu, HI, USA, July 2017.

[12] F. Zhang, H. Zhao, W. Ying, Q. Liu, A. Noel et al., “Human face sketch to RGB image with edge optimization and generative adversarial networks,” Intelligent Automation & Soft Computing, vol. 26, no. 6, pp. 1391–1401, 2020.

[13] K. Fu, J. Peng, H. Zhang, X. Wang, and F. Jiang, “Image super-resolution based on generative adversarial networks: a brief review,” Computers, Materials & Continua, vol. 64, no. 3, pp. 1977–1997, 2020.

[14] M. Zhao, X. Liu, X. Yao, and K. He, “Better visual image super-resolution with laplacian pyramid of generative adversarial networks,” Computers, Materials & Continua, vol. 64, no. 3, pp. 1601–1614, 2020.

[15] S. Deivalakshmi, P. Palanisamy, and X. Z. Gao, “Balanced GHM multiwavelet transform based contrast enhancement technique for dark images using dynamic stochastic resonance,” Intelligent Automation and Soft Computing, vol. 25, no. 3, pp. 459–471, 2019.

[16] J. Li, L. Hao, G. Cui, Y. Kang, Y. Hu, and Y. Zhou, “Gacnet: a generative adversarial capsule network for regional epistaxial traffic flow prediction,” Computers, Materials & Continua, vol. 64, no. 2, pp. 925–940, 2020.

[17] J. Wang, Y. Yang, T. Wang, R. Simon Sherratt, and J. Zhang, “Big data service architecture: a survey,” Journal of Internet Technology, vol. 21, no. 2, pp. 393–405, 2020.

[18] G. Wang, C. Sun, and A. Somwya, “Attentive feature refinement network for single rainy image restoration,” IEEE Transactions on Image Processing, vol. 30, no. 2021, pp. 3734–3747.

[19] S. Li, W. Ren, F. Wang et al., “A comprehensive benchmark analysis of single image deraining: current challenges and future perspectives,” International Journal of Computer Vision, vol. 129, no. 4, pp. 1301–1322, 2021.

[20] C. Wang, Y. Wu, Y. Cai et al., “Single image deraining via deep pyramid network with spatial contextual information aggregation,” Applied Intelligence, vol. 50, no. 5, pp. 1437–1447, 2020.

[21] L. Cerkezi and C. Topal, “Towards more discriminative features for texture recognition,” Pattern Recognition, vol. 107, Article ID 107473, 2020.

[22] Y. Li, R. T. Tan, and X. Guo, “Rain streak removal using layer priors,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2736–2744, Las Vegas, NV, USA, June 2016.

[23] X. Jin, Z. Chen, and J. Lin, “A decomposed dual-cross generative adversarial network for image rain removal,” in Proceedings of the British Machine Vision Conference (BMVC’18), vol. 119, Newcastle, UK, September 2018.

[24] T. Ojala, M. Pietikainen, and D. Harwood, “Performance evaluation of texture measures with classification based on Kull back discrimination of distributions,” in Proceedings of the IAFR International Conference on Pattern Recognition, IEEE, Quebec, Canada, August 2002.

[25] N. Hor and S. Fekri-Ershad, “Image retrieval approach based on local texture information derived from predefined patterns and spatial domain information,” International Journal of Computer Science Engineering, vol. 8, pp. 246–254, 2019.

[26] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Proceedings of the IEEE Computer Society Conference Computer Vision and Pattern Recognition (CVPR’05), vol. 1, pp. 886–893, SanDiego, CA, USA, June 2005.

[27] A. Khan, A. Chefranov, and H. Demirel, “Image-level structure recognition using image features, templates, and ensemble of classifiers,” Symmetry, vol. 12, no. 7, p. 1072, 2020.

[28] S. Tulyakov, M. Y. Liu, and X. Yang, “MoCoGAN: decomposing motion and content for video generation,” 2017, https://arxiv.org/abs/1707.04993.

[29] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising,” IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3142–3155, 2017.

[30] Z. C. Zhu, B. Li, and S. Fang, “A restoration algorithm for rain video image,” Journal of Hefei University of Technology (Natural Science), vol. 34, pp. 1011–1014, 2011.

[31] R. Dubolia, R. Singh, and S. S. Bhadoria, “Digital image watermarking by using discrete wavelet transform and discrete cosine transform and comparison based on PSNR,” in Proceedings of the International Conference on Communication Systems and Network Technologies, pp. 593–596, IEEE, Katra, Jammu, India, June 2011.