FastDerainNet: A Deep Learning Algorithm for Single Image Deraining

XIUWEN WANG¹, ZHIWEI LI¹,², HONGTAO SHAN¹, ZHIYUAN TIAN¹, YUANHONG REN², AND WUNENG ZHOU²
¹College of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China
²College of Information Science and Technology, Donghua University, Shanghai 200051, China
Corresponding authors: Zhiwei Li (zhiwei.li@sues.edu.cn) and Wuneng Zhou (wnzhou@dhu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61705127, in part by the Shanghai Natural Science Foundation under Grant 19ZR1421600, and in part by the Degree Construction Project of Detection Technology and Automation Devices, Shanghai University of Engineering Science, under Grant 19XXK003.

ABSTRACT Existing neural network-based methods for de-raining single images exhibit dissatisfactory results owing to the inefficient propagation of features when objects with sizes and shapes similar to those of rain streaks are present in images. Furthermore, existing methods do not consider that the abundant information included in rain streaked images could interfere with the training process. To overcome these limitations, in this paper, we propose a deep residual learning algorithm called FastDerainNet for removing rain streaks from single images. We design a deep convolutional neural network architecture, based on a deep residual network called the share-source residual module (SSRM), by substituting the origins of all shortcut connections for one point. To further improve the de-raining performance, we adopt the SSRM as the parameter layers in FastDerainNet and use image decomposition to modify the loss function. Finally, we train FastDerainNet on a synthetic dataset. By learning the residual mapping between rainy and clean image detail layers, it is able to reduce the mapping range and simplify the training process. Experiments on both synthetic and real-world images demonstrate that the proposed method achieves increased performance with regard to de-raining, in addition to preserving original details, in comparison with other state-of-the-art methods.

INDEX TERMS Deep residual learning, rain streak removal, FastDerainNet, share-source residual module.

I. INTRODUCTION Pictures taken in the rain are more likely to lose high-frequency information because of the existence of rain streaks; thus, the quality is degraded. However, most high-level computer vision tasks, such as image classification [1], [2], object detection [3], [4], and image segmentation [5] require high quality images. Therefore, rain streak removal has become essential in the field of computer vision. Fig. 1 exhibits an example of single image rain streak removal.

In the last few years, numerous methods have been proposed for single image rain streak removal. These methods either employ models with different priors [6]–[8], or sparse dictionary learning [9], [10]. Although these models have achieved extensive progress in some cases, there is still substantial room for improvement.

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Asikuzzaman.

Owing to powerful feature representation [1], several deep learning models have been proposed for removing rain streaks [11]–[22]. However, most existing methods suffer from two significant limitations. First, some methods [14]–[16] directly apply deep convolutional neural networks to eliminate the complex influence of rain streaks. However, actual results are dissatisfactory, as they do not take into consideration that extraneous information included
in images could interfere with the training process. Second, when an object’s shape and size are similar to those of rain streaks, such deep learning methods [17], [18] are unable to effectively propagate the features, which can generate over-smoothed results.

In view of the above issues and inspired by a residual network (ResNet) [23], [24] and a dense convolutional network (DenseNet) [25], we designed an SSRM specifically for de-raining. Then, we combined the SSRM with image processing domain knowledge using a structure similar to that of DerainNet [12], to create a deep residual learning algorithm called FastDerainNet, as shown in Fig. 2.

The main contributions of our work are summarized as follows:

1) We propose a modified residual architecture, the SSRM, in which all shortcut connections share the same starting point. This change from the original ResNet can accelerate convergence and help to obtain the de-rained result more rapidly.

2) We introduce FastDerainNet, which directly learns the nonlinear relationship between the detail layers of rainy and clean images. FastDerainNet decomposes the image and trains the SSRM on the high-frequency detail layer, thereby reducing the mapping range. This further improves de-raining performance.

3) Further, using real-world clean images as the ground truth, we synthesize a dataset to train our network. Experimental results demonstrate that the proposed method can effectively and efficiently perform rain removal.

II. METHODOLOGY

In this section, we describe and analyze the framework of the proposed FastDerainNet from four aspects: the SSRM architecture, image processing, negative residual mapping [13], and the loss function.

Fig. 3 presents the ground truth image and rainy image of a flower, along with de-rained images produced by different components of the proposed algorithm, and the proposed algorithm itself. We can express the similarity between each image and the ground image as the structural similarity index (SSIM) [28], and use the peak signal-to-noise ratio (PSNR) [29] as another quality metric. These results, shown in Table 1, demonstrate the effect of each component of the proposed algorithm.

| SSIM/PSNR | Ground truth | Rainy image | SSRM | Image Processing | SSRM + Neg-mapping | FastDerainNet |
|-----------|--------------|-------------|------|------------------|--------------------|---------------|
| 0.69/21.22 | 0.79/27.96   | 0.78/27.73  | 0.82/27.57 | 0.91/32.13 |

A. SSRM ARCHITECTURE

Owing to its shortcut connections [23], ResNet significantly increases image classification performance. However, it requires a large number of calculations because these identity mappings [24] are connected one by one. Thus, we introduce the SSRM to further improve the information flow between layers and accelerate convergence. Specifically, this method will use the same source and identify a variable in advance, to optimize the calculation of each residual block.
As shown in Fig. 4, both ResNet and the SSRM with \( N \) layers include \((N-2)/2\) residual blocks and shortcut connections when compared with a standard convolutional neural network (CNN). In ResNet, all residual blocks are assembled end-to-end; in other words, these shortcut connections do not strongly depend on each other, even though they are trained jointly. However, the integrity and adaptability of the foundation in the SSRM are strengthened by making all shortcut connections share the same origin. Hence, all residual blocks are different but closely linked.

Through these shortcut connections, both feedforward and feedback signals can be directly transported. Traditional feedforward architectures pass on information that needs to be preserved, whereas ResNet propagates information between layers by building additive identity mappings. In this sense, the SSRM preserves the above two characteristics. Thus, compared with a CNN, the SSRM improves the flow of information and strengthens feature propagation. When compared with ResNet, the SSRM can better ensure the integrity of information and strengthens feature propagation. Thus, the integrity and adaptability of the foundation in the SSRM are strengthened by making all shortcut connections share the same origin. Hence, all residual blocks are different but closely linked.

As shown in Fig. 4 (c), if we set \( I_0 \) as the input, \( I_{in} \) and \( I_{out} \) respectively represent the input and output of the SSRM in the \( n \)th layer. Following this, the structure of the SSRM can be described as follows:

\[
\begin{align*}
I_{in}^1 &= I_{in}^0 \\
I_{in}^2 &= \sigma(BN(W^1 \ast I_{in}^1 + b^1)), \\
I_{out}^1 &= I_{in}^1 \\
I_{out}^2 &= \sigma(BN(W^2 \ast I_{in}^2 + b^2)), \\
I_{out}^3 &= I_{out}^2 \\
I_{out}^4 &= \sigma(BN(W^3 \ast I_{in}^3 + b^3)).
\end{align*}
\]  

(3)

B. IMAGE PROCESSING

Based on the literature [12], the rainy image can be considered to consist of two parts: the base layer and the detail layer, which can be expressed mathematically as

\[
R = R_{base} + R_{detail}.
\]  

(6)

where \( R \) is the rainy image and the subscripts ‘detail’ and ‘base’ represent the detail and base layers, respectively. Because there are only rain streaks and partial object structures remaining in \( R_{detail} \), the detail layer is sparser than the rainy image. Thus, we train the SSRM on the detail layer instead of the image domain.

Instead of randomly choosing filters, we always employ the same principle that DerainNet employs [12] to reduce the solution space by compressing the mapping range. As shown in Fig. 2, we choose two filters to minimize \( R_{neg-mapping} \), which are formulated as

\[
R_{neg-mapping} = C - R_{base} - R_{detail}.
\]  

(7)

First, via a low-pass filter, we obtain the optimized \( R_{base}^* \) which is formulated as

\[
R_{base}^* = \arg \min_{R_{base}}(C - R_{base}).
\]  

(8)

Likewise, based on the known \( R_{base}^* \), we obtain \( R_{detail}^0 \) via a high-pass filter

\[
R_{detail}^0 = \arg \min_{R_{detail}}((C - R_{base}^*) - R_{detail}).
\]  

(9)

Finally, according to Eqs. (7)–(9), we determine the optimal \( R_{neg-mapping}^1 \), which greatly simplifies the training process.

To measure the similarity between two images, the structural similarity index (SSIM) [28] and peak signal-to-noise ratio (PSNR) [29] are used for quantitative evaluation.
In addition, we adopt non-reference metrics, a blind/referenc- celess image spatial quality evaluator (BRISQUE) [38], and an integrated local naturalness image quality evaluator (IL-NIQE) [39], which measure the visual quality in different ways. We decompose the image into a high-frequency part \( R_{\text{detail}} \) and a low-frequency part \( R_{\text{base}} \) using a bilateral filter [30] and a guided filter [31], respectively. These results are shown in Table 2.

**TABLE 2.** The results for several filtering methods.

| Image       | Bilateral filter + Gaussian filter [36] | Bilateral filter + Guided filter | Bilateral filter + Guided filter |
|-------------|----------------------------------------|---------------------------------|---------------------------------|
| Deer        | 0.674/26.985                           | 0.803/32.562                   | 0.800/32.547                   |
| Tiger       | 0.755/29.635                           | 0.858/33.959                   | 0.850/33.789                   |
| 200 images  | 0.744/28.334                           | 0.832/33.520                   | 0.835/33.783                   |

**C. NEGATIVE RESIDUAL MAPPING**

When compared with clean image \( C \), the pixel values of rainy image \( R \) are higher because rain streaks tend to appear white in images. Thus, most \( C - R \) values tend to be negative; this is known as “negative residual mapping.”

As shown in Fig. 2, \( R_{\text{neg-mapping1}} \) and \( R_{\text{neg-mapping2}} \), two negative residual mappings, are built to improve the efficiency of image decomposition. First, by learning the residual values between output image \( C \) and the base layer \( R_{\text{base}} \), we obtain the de-rained residual layer \( R_{\text{neg-mapping2}} \), which can be expressed as

\[
R_{\text{neg-mapping2}} = C - R_{\text{base}}, \quad (10)
\]

Following this, by learning the residual between \( R_{\text{neg-mapping2}} \) and detail layer \( R_{\text{detail}} \), we obtain the detail residual layer \( R_{\text{neg-mapping1}} \), formulated as,

\[
R_{\text{neg-mapping1}} = R_{\text{neg-mapping2}} - R_{\text{detail}}, \quad (11)
\]

\[
R_{\text{neg-mapping1}} = f_w^b(R_{\text{detail}}), \quad (12)
\]

where \( R_{\text{neg-mapping1}} \) represents the output of the SSRM, and \( f_w^b(\cdot) \) denotes the SSRM.

**D. OBJECTIVE FUNCTION**

According to Eqs. (10)–(12), the objective function of FastDerainNet can be defined as the mean square error (MSE), which is formulated as

\[
L(W, b) = \frac{1}{M} \sum_{m=1}^{M} \| f_W^b(mR_{\text{detail}}) + mR_{\text{detail}} - (mC - mR_{\text{base}}) \|_F^2 .
\]

where \( M \) is the total number of training samples, \( m \) indexes the image, and \( \| \cdot \|_F \) is the Frobenius norm.

**III. EXPERIMENT**

**A. DATASET**

As it is difficult to obtain a ground truth corresponding to real-world rainy images, we collected 2000 clean images and synthesized 12,000 rainy images. For example, Fig. 5 (a) is used as the ground truth to generate six rainy images with three directions and two magnitudes. To improve the reliability of the model, we took various measures to perfect our synthetic data. First, to create the greatest number of training samples possible, we randomly selected 9000 images to generate 3 million 64 × 64 rainy and clean patch pairs with a mini patch size of 20. The remaining image pairs were used for validation and evaluation.

Furthermore, we adopted partially synthetic data to optimize the data sets. Specifically, we used real data as the initial sample, with the synthetic data filled. Therefore, the model differs between various executions because of the differences in the components of the data set. Finally, by comparing the metrics, we found that the difference between the models is small, which proves that our synthetic dataset can be generalized to the real world.

**B. PARAMETER SETTINGS**

We set the network depth to \( N = 18 \) and used stochastic gradient descent (SGD) [32] to minimize Eq. (13) with a weight decay of \( 10^{-10} \) and momentum of 0.9. The learning rate was initialized at 0.2 and multiplied by 0.998 per 100 iterations, with training terminating at 120,000 iterations. We set filter sizes to \( 3 \times 3 \), and the channel was set to 3; filter numbers and more concrete configurations are presented in Table 3. We did not use pooling with respect to possible adverse impacts on the accuracy of the network, even where overfitting and dimensionality reduction were anticipated.

**C. RESULTS ON SYNTHETIC IMAGES**

Fig. 6 shows visual comparisons for four synthetic rainy images; two regions of interest are selected to highlight the effects. Notably, DerainNet fails to remove the rain streaks in heavy rain, whereas joint rain detection and removal (JORDER) [13] and the density-aware single image de-raining multi-stream dense network (DID-MDN) [15] are able to remove most of the heavy rain streaks.
while also generating obvious artifacts. However, both recurrent squeeze-and-excitation context aggregation network (RESCAN) [22] and FastDerainNet efficiently distinguish most of the rain streaks and achieve better visual effects. To further illustrate this improvement, Table 4 lists the calculated metrics for each of the three methods, using PSNR and SSIM for quantitative evaluation. Our results for the proposed method are both subjectively and objectively superior to those of the other methods evaluated.

D. RESULTS ON REAL-WORLD IMAGES
Since a ground truth corresponding to real-world rainy images was not available, it is impossible to quantitatively compare the results. As shown in Fig. 7, JORDER still contains rain streaks, and the results from DerainNet contain obvious unprocessed artifacts. Furthermore, DID-MDN, tends to blur image details and generate over-smoothed images. In contrast, RESCAN and FastDerainNet produce far fewer artifacts, thereby preserving more textural detail. However, the results of RESCAN seem darker, which could

### TABLE 3. The architecture and parameters of FastDerainNet.

| Five main parts | Filter number | Input shape | Output shape | Use pool |
|-----------------|---------------|-------------|--------------|----------|
| Input           |               | [20,64,64,3]| [20,64,64,3]| no       |
| Conv 1          | 32            | [20,64,64,3]| [20,64,64,32]| no       |
| Conv 1–17       | 32            | [20,64,64,32]| [20,64,64,32]| no       |
| Conv 18         | 3             | [20,64,64,32]| [20,64,64,3]| no       |
| Output          |               |             | [20,64,64,3]|          |

### TABLE 4. De-rain performance of six different methods on synthesized rainy images.

| Image         | Ground truth | Rainy images | DerainNet method [12] | DID-MDN method [15] | JORDER method [13] | RESCAN method [22] | FastDerainNet |
|---------------|--------------|--------------|------------------------|---------------------|--------------------|--------------------|---------------|
| Deer          | 1/60         | 0.6377/19.3023 | 0.8640/27.1627         | 0.9210/28.6207      | 0.7773/25.1369     | 0.8549/24.4712   | 0.9598/33.2290 |
| Tiger         | 1/60         | 0.7575/21.0911 | 0.8634/26.7336         | 0.8876/27.9792      | 0.8751/23.6896     | 0.9273/30.2801   | 0.9166/28.6311 |
| Flower1       | 1/60         | 0.4463/14.1239 | 0.6576/22.6345         | 0.7179/25.9456      | 0.5850/19.1759     | 0.7634/18.5939   | 0.8162/24.3861  |
| Flower2       | 1/60         | 0.6483/19.1420 | 0.8226/27.6346         | 0.8619/22.9737      | 0.8255/24.5142     | 0.8266/18.2555   | 0.9334/32.6486  |
| 200 images    | 1/60         | 0.5467/16.3192 | 0.7749/24.2873         | 0.8417/25.1946      | 0.7938/23.2543     | 0.8711/24.6537   | 0.8836/27.8279  |
produce negative effects on visual perception. The BRISQUE values are shown in Table 5. We recruited participants to rate each group of results. For each result, 30 participants were required to respond to each of the two questions shown in Fig. 8, using a Likert scale [40] ranging from poor to excellent. Results are shown in Fig. 8, where each subfigure shows five rating distributions for the methods on a specific question. The distributions across methods show that FastDerainNet is preferred by human observers; our method received more “excellent” and fewer “poor” ratings compared with other methods.

**E. COMPUTATIONAL COMPLEXITY**

Table 6 compares the runtimes of the different state-of-the-art methods. All methods were implemented in MATLAB and pycharm on a GPU, following the original settings of all methods.
the released codes. Note that the DID-MDN method [15] requires the input image size to be fixed to $512 \times 512$; therefore, we tested only $512 \times 512$-sized images for all methods. To balance the trade-off between performance and computation efficiency, we chose 16 parameter layers for the above experiments.

It was observed that DerainNet [12] and RESCAN [22] consume a substantial amount of time because of a complex optimization process. The computation time of JORDER is limited by redundant network parameters [13]; regardless, we found it to be faster than RESCAN. The batch normalization layers in DerainNet [12] cause this method to consume more test time than other methods, although it uses fewer network parameters. The average running time and parameter number results show that FastDerainNet is an impressive method quantitatively, in spite of employing a light-weight framework.

### IV. CONCLUSION

In this paper, we proposed a novel end-to-end network named FastDerainNet to remove rain streaks from single images. To enhance the effects, we designed a novel residual architecture, called the SSRM. Compared with other CNNs, the SSRM shared more information between layers, thereby making more relevant information available. In addition, by using the SSRM on high-frequency components, FastDerainNet directly learned the residuals between rainy and clean images. Experiments on both synthetic and real-world rainy images demonstrated that our method achieves superior SSIM, PSNR, BRISQUE, and NIQE performance compared with state-of-the-art methods.

Finally, it should be noted that, although we proposed FastDerainNet for de-raining problems, this method can be applied equally to other image-to-image transformation tasks, such as denoising [33], dehazing [34], and de-blurring [35].

### REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” Commun. ACM, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[3] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/tpami.2016.2577031.

[4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, pp. 1627–1645, Sep. 2010, doi:10.1109/tpami.2009.167.

[5] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2015, pp. 3431–3440, doi: 10.1109/cvpr.2015.7298965.

[6] Y. Wang, S. Liu, C. Chen, and B. Zeng, “A hierarchical approach for rain or snow removing in a single color image,” IEEE Trans. Image Process., vol. 26, no. 8, pp. 3936–3950, Aug. 2017, doi: 10.1109/tip.2017.2708502.

[7] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, “Rain removal using layer priors,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2736–2744, doi: 10.1109/cvpr.2016.299.

[8] Y. Luo, J. Zhu, J. Ling, and E. Wu, “Fast removal of rain streaks from a single image via a shape prior,” IEEE Access, vol. 6, pp. 60069–60078, Nov. 2018, doi: 10.1109/access.2018.2875171.

[9] 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, doi: 10.1109/tip.2011.2179057.

[10] Y. Luo, Y. Xu, and H. Ji, “Removing rain from a single image via discriminative sparse coding,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 3397–3405, doi: 10.1109/iccv.2015.388.
X. Wang et al.: FastDerainNet: A Deep Learning Algorithm for Single Image Deraining

[11] P. Mu, J. Chen, R. Liu, X. Fan, and Z. Luo, “Learning bilevel layer priors for single image rain streaks removal,” IEEE Signal Process. Lett., vol. 26, no. 2, pp. 307–311, Feb. 2019, doi: 10.1109/7892277.

[12] 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, doi: 10.1109/tip.2017.2691902.

[13] 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), Jul. 2017, pp. 1357–1366, doi: 10.1109/cvpr.2017.183.

[14] 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. 3855–3863, doi: 10.1109/cvpr.2017.186.

[15] H. Zhang and V. M. Patel, “Density-aware single image de-raining using a multi-stream dense network,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 695–704, doi: 10.1109/cvpr.2018.00079.

[16] X. Fu, R. Zhuge, H. Li, S. Song, F. Jiang, and M. Xu, “Single image rain removal via a simplified residual dense network,” IEEE Access, vol. 6, pp. 66522–66535, Nov. 2018, doi: 10.1109/access.2018.2879300.

[17] B. Cai, X. Xu, K. Jia, C. Qin, and D. Tao, “DehazeNet: An end-to-end system for single image haze removal,” IEEE Trans. Image Process., vol. 25, no. 11, pp. 5187–5198, Nov. 2016, doi: 10.1109/tip.2016.2598681.

[18] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, “Scale-recurrent network for deep image deblurring,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8174–8182, doi: 10.1109/cvpr.2018.00853.

[19] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” IEEE Trans. Signal Process., vol. 50, no. 2, pp. 174–188, Aug. 2002, doi: 10.1109/78.978374.

[20] T. Dal, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, “Second-order attention network for single image super-resolution,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 11057–11066, doi: 10.1109/cvpr.2019.01132.

[21] A. Mittal, A. K. Moothy, and A. C. Bovik, “Blind/referenceless image spatial quality evaluator,” in Proc. Conf. Rec. Asilomar Conf. Signals, Syst. Comput., Nov. 2011, pp. 723–727.

[22] L. Zhang, L. Zhang, and A. C. Bovik, “A feature-enriched completely blind image quality evaluator,” IEEE Trans. Image Process., vol. 24, no. 8, pp. 2579–2591, Aug. 2015.

[23] R. Wang, Q. Zhang, C.-W. Fu, X. Shen, W.-S. Zheng, and J. Jia, “Under-exposed photo enhancement using deep illumination estimation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 6849–6857, doi: 10.1109/cvpr.2019.00701.

[24] L. Song, C. Wang, L. Zhang, B. Du, Q. Zhang, C. Huang, and X. Wang, “Unsupervised domain adaptive re-identification: Theory and practice,” 2018, arXiv:1807.11334. Online. Available: http://arxiv.org/abs/1807.11334

XIUWEN WANG was born in 1997. He is currently pursuing the M.S. degree with the Shanghai University of Engineering Science. His research interests include image processing and machine learning.

ZHWEI LI received the M.S. degree in electronic science and technology from Wuhan University, in 2006, and the Ph.D. degree from the Huazhong University of Science and Technology, in 2016. He is currently an Associate Professor with the Shanghai University of Engineering Science. His research interests include intelligent theory and control, image processing, and optoelectronics.

HONGTAO SHAN received the M.S. degree in electric engineering and the Ph.D. degree from the Huazhong University of Science and Technology, in 2005 and 2009, respectively. She is currently an Associate Professor with the Shanghai University of Engineering Science. Her research interests include visual inspection and image processing.
ZHIYUAN TIAN was born in 1996. He is currently pursuing the M.S. degree with the Shanghai University of Engineering Science. His research interests include control systems, image processing, and machine learning.

YUANHONG REN received the B.S. degree in information and computing science from the Wuhan University of Science and Technology and the M.S. degree in computer application technology from the Wuhan University of Technology. Her research interests include image processing and intelligent theory and control.

WUNENG ZHOU received the M.S. degree from Central China Normal University, in 1982, and the Ph.D. degree in control science and engineering from Zhejiang University, in 2005. He is currently a Professor with Donghua University. His research interests include system theory and control theory, control engineering, and robust control theory and application.

* * *