Utilizing Two-phase Processing with FBLS for Single Image deraining

Xiao Lin, Lizhuang Ma, Bin Sheng, Zhi-Jie Wang, Wansheng Chen

***Abstract*—Rain removal from a single image is a challenging problem and has attracted much attention in recent years. In this paper, we revisit the single image deraining problem, and present a novel solution. The central idea of our solution is to merge the merits of two-phase processing methods and the Fuzzy Broad Learning System (FBLS). Specifically, our solution first uses the dehazing algorithm to preprocess the input rainy image and separates it into the detail layer and the base layer. After that, it puts the Y-channel image of the detail layer into the FBLS to obtain the derained Y channel image, which is then combined with the Cb and Cr channel images to obtain the derained detail layer. Later, it fuses the derained detail layer and the base layer to get a preliminary derained image. Finally, it superimposes the details extracted from the dehazed image with some transparency on the preliminary result, obtaining the final result. Experimental results based on both real and synthetic rainy images demonstrate that our proposed solution can outperform several state-of-the-art algorithms, while it consumes much less running time and training time, compared against the competitors.**

***Index Terms*—Single image rain removal, computer vision, machine learning, image processing.**

I. Introduction

In recent years, single image dehazing and deraining are very hot topics in the fields of multimedia, computer vision, and image processing [1], [2], [3], [4], [5], [6], [7], [8]. In this paper, we focus on discussing the problem of single image deraining. The rain in the image can be roughly divided into two cases: (i) rain streaks near to the camera lens, which can be considered as noise in the image; and (ii) rain in a long distance, which looks like a translucent veil or fog. Under rainy conditions, the impact of rain streaks on images and video is often undesirable. In addition to a subjective degradation, the effects of rain can also severely affect the performance of many multimedia processing systems, e.g., content-based image retrieval [9], image enhancement methods [10], [11], surveillance systems [12], [13]. Fig. 1 shows examples of rain removal. From the figure, one can easily see that there are great differences between images before and after removing rain streaks. This implies that rain removal is great important, and thus has attracted much attention in past years.

In the literature, previous works related to rain removal can be roughly divided into two branches: Video-based methods [14], [15], which remove rain streaks from video; and single-image-based methods [13], [16], [17], which remove rain streaks from a single rainy image. In this paper, we are interested in the latter. In past years, many methods/techniques were proposed for rain removal from a single image, such as sparse codes [18], layer priors [19], guided L0 smoothing filter [20], anisotropic filter [21]. Generally, recent methods can be classified into three categories [22]. The first category is simply filtering-based where a nonlocal mean filter or guided filter is often used [23], [20], [24]. These methods run fast while they can hardly produce a satisfactory performance consistently. The second category builds models for rain streaks [25], [18], [19]. These models can discriminate rain streaks from the background. However, it often happens that some details of the image will be mistreated as rain streaks. The third category, which seems more reasonable, is to form a two-phase processing [26], [27], [28]. Specifically, a well-designed filtering is first used to decompose a rainy image into the low-frequency part and high-frequency part; the low-frequency part can be made free of rain as much as possible, the high-frequency part can be used to further extract the image's details to be added back into the low-frequency part.

On the other hand, in the existing works (including the works mentioned above), some researches [29], [30], [12], [31] employed the deep learning technology, owing to its success in many other applications, such as visual recognition [32], semantic segmentation [33], natural language processing [34], [35], bioinformatics [36]. Nevertheless, it is well known that deep learning technique bears also some issues/limitations, such as high hardware requirement, long training time, and too many hyper-parameters. Recently, the fuzzy broad learning system (FBLS) [37], [38] has been shown useful for regression and classification. An obvious strong point is that, unlike deep learning models which consist of the stack of neuron layers, the FBLS has only one neuron layer and it does not need to stack layers in deep, but only need to expand in broad. In addition, its hyper-parameters are far less than that of deep learning based models, and so its training process is very fast.

Motivated by the above facts, in this paper we attempt to integrate the two-phase processing with FBLS for single image deraining task. Specifically, we first use the dehazing algorithm

X. Lin is with the Department of Computer Science, Shanghai Normal University, Shanghai 200234, China; and also with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (Email: Lin6008@126.com).

L. Ma is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China; and also with the School of Computer Science and Technology, East China Normal University, Shanghai 200241, China (Email: ma-lz@cs.sjtu.edu.cn).

B. Sheng is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (Email: shengbin@sjtu.edu.cn).

Z.-J Wang is with the College of Computer Science, Chongqing University, Chongqing 400044, China, also with the School of Data and Computer Science, Sun Yat-Sen University, Guangzhou 510006, China (e-mail: cszjwang@yahoo.com).

W. Chen is with the Department of Computer Science and Engineering, Shanghai Normal University, Shanghai 200234, China (Email: 1246523423@qq.com).

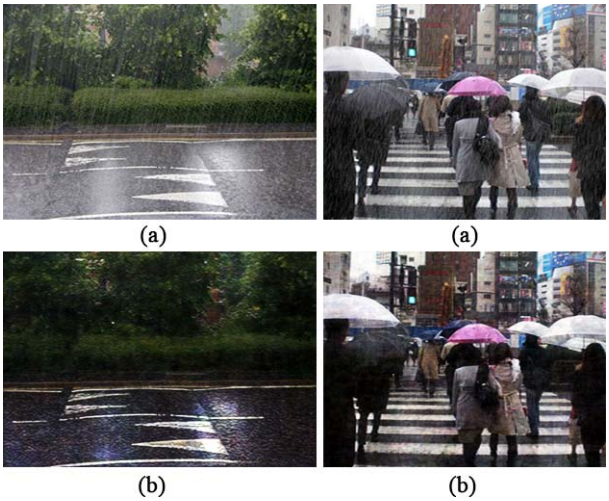


Fig. 1: Single image rain removal example. (a) Rainy image. (b) image after deraining.

based on dark channel prior [39] to process the input rainy image. This makes the misty rain in the distance removed and highlights the rain streaks hidden in the "fog". Then, we use a filter to obtain the detail layer and the base layer, respectively. To reduce the complexity of computation and/or training, we convert the RGB color space of the detail layer to the YCbCr color space, and puts Y channel image of YCbCr color space into FBLS for obtaining the derained Y-channel image (notice: in the training phase, it puts Y channel image of YCbCr color space and ground truth image into FBLS for training). After that, we combine the derained Y-channel image with the previous Cb and Cr channel images, in order to get the derained three-channel detail layer. Next, the processed detail layer is fused with the base layer to get a preliminary derained image. To strengthen the quality of the preliminary result, we further extract details from the dehazed image, and the extracted details are superimposed on the preliminary result, obtaining the enhanced derained image (i.e., the final result). To summarize, the main contributions of this paper are as follows:

- We propose a novel solution for rain removal from a single image. Our solution combines the two-phase processing with FBLS seamlessly. To our knowledge, this could be the first attempt to combine two-phase processing with FBLS for single image deraining task.
- We conduct the empirical study based on both real and synthetic rainy images. The experimental results show us that the proposed solution can beat several state-of-the-art algorithms, in terms of solution quality. Particularly, our solution has much fewer training and running time, compared against several strong competitors.

The rest of the paper is organized as follows. In next section, we review related work. Section III presents our solution in detail. The experimental results and discussions are covered in Section IV. Finally, we conclude this paper in Section V.

II. Related Work

In the existing literature, previous works related to rain removal are mainly in two branches: (i) rain removal from a video, and (ii) rain removal from a single image. For category (i), video-based methods are extensively studied in past years [14], [40], [41]. In addition, as pointed out in [12], rain can be more easily identified and removed using inter-frame information [40], [41]. Many of these methods work well, but are significantly aided by the temporal content of video. Unlike the task of removing rain from a video, the single image deraining task is significantly more challenging, since much less information is available for detection and removing rain. In what follows, we focus more of our attention on reviewing previous works in this branch.

In the past years, there are already some works addressing the problem of rain removal from a single image [23], [20], [24], [42], [30], [25], [18], [19], [26], [27], [28]. As mentioned in [22], existing methods can be generally classified into three categories. **(1)** The methods based on simple filtering [23], [20], [24], in which a nonlocal mean filter or guided filter is often used. For example, Xu *et al.* [24] designed a rain-free guidance method to remove rain from a single image. Kim *et al.* [23] detected the rain streak regions and removed rain in a single image by nonlocal means filtering. Wang *et al.* [21] used the anisotropic filter to remove rain streaks. More recently, Ding *et al.* [20] removed rain streaks from a single image by combining L0 gradient minimization with a guided filter. A common feature of this line of methods is that, they run fast while they can hardly produce a satisfactory performance consistently (either the output image is left over with some rain streaks, or quite a few image's details are lost so that the output image becomes blurred). **(2)** The model-based methods [25], [18], [19], which usually build models for rain streaks. For example, Chen *et al.* [25] found that rain streaks always appear similar patterns, and so they proposed a low-rank appearance model to remove rain streaks from a single image. Luo *et al.* [18] suggested to use the discriminative sparse codes that are based on the non-linear generative model of rainy image. Li *et al.* [19] made use of some patch-based prior model for both the background layer and rainy layer for the rain removal task. This line of methods or models can discriminate rain streaks from the background. However, it often happens that some details of the image will be mistreated as rain streaks. **(3)** The two-phase processing based methods [26], [27], [28], which seem to be more reasonable. Specifically, they usually use a well-designed filtering to decompose a rainy image into the low-frequency part and high-frequency part; the low-frequency part can be made free of rain as much as possible, the high-frequency part is used to further extract image's details. For example, the authors in [27], [26] proposed a single-frame-based rain removal framework by properly formulating the rain removal task as an image decomposition problem, based on morphological component analysis. Later, Chen *et al.* [28] also proposed a single-color-image-based rain removal framework, but they are based on sparse representation. Later, Wang *et al.* [22] further advanced this line of methods by combining image decomposition and dictionary learning. To some extent, our

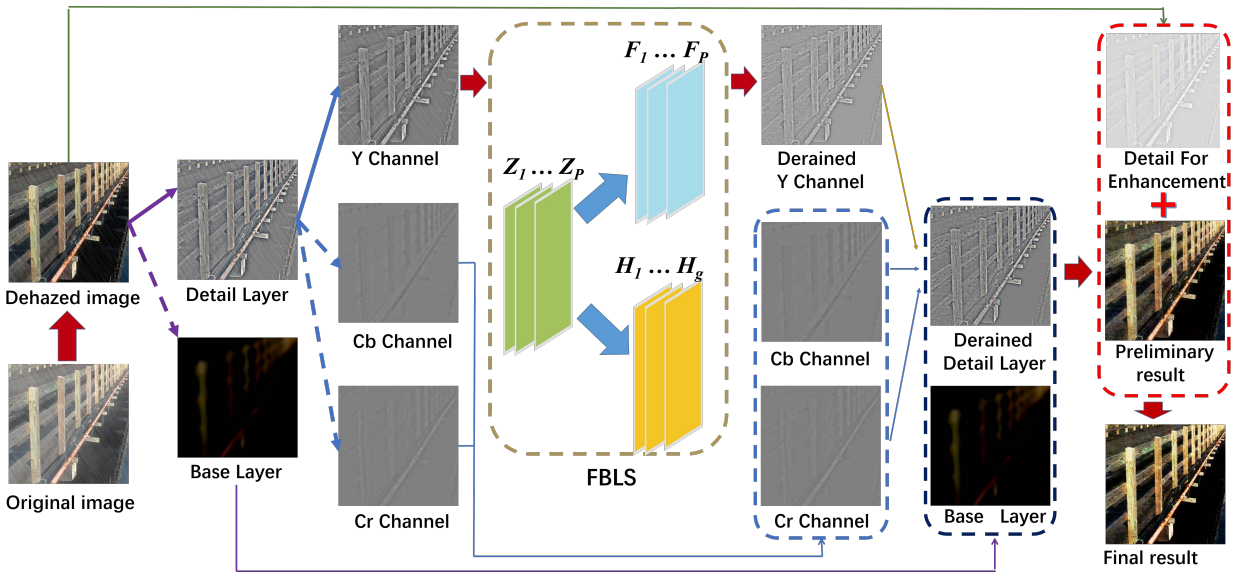


Fig. 2: The overall framework of our solution.

paper is similar to the methods in category (3), i.e., two-phase processing based methods. This is mainly because we also separate the image into the base and detail layers, using the low and high pass filtering, respectively. Nevertheless, our work is different from these works in several points at least: (i) we use the dehazing technique to remove fogs before separating the image; (ii) we employ the fuzzy broad learning system to assist the deraining task; and (iii) we extract the details from the dehazed rainy image to further strengthen the result.

On the other hand, in the existing works (including the works mentioned above), some works (e.g., [29], [30], [12], [43]) employed the deep learning technology, owing to its success in many other applications such as recommendation [44], [45], visual recognition [32], semantic segmentation [33], [46], saliency detection [47], [48], moving prediction [49], [50], [51], [52], etc. Meanwhile, it is well known that deep learning technique bears also some issues/limitations, such as high hardware requirement, long training time, and too many hyper-parameters. To alleviate these issues, the fuzzy broad learning system (FBLS) [37], [38] has been extensive studied recently. It combines/integrates the Takagi-Sugeno (TS) fuzzy system [53] on basis of broad learning system (BLS) [54]. The BLS is a framework that can replace the structure of deep learning. The strong points of BLS are that, it can alleviate the troubles/issues (e.g., long training time) caused by a large number of parameters in deep learning based models. Unlike deep learning models, which consist of the stack of neuron layers, the BLS has only one neuron layer, which are composed of feature nodes and enhancement nodes. In addition, its hyper-parameters are far less than that of deep learning based models, and so its training process is very fast. To fuse TS fuzzy system with BLS, the FBLS replaces the feature nodes in BLS with a group of TS fuzzy subsystems, and then the intermediate outputs (generated by all fuzzy subsystems) are sent to the enhancement nodes as a vector connection for further nonlinear transformation. The final output is generated by combing the outputs from both the fuzzy subsystem and the enhancement nodes. Since the FBLS saves the adjustment process of sparse autoencoder in BLS, it reduces the complexity of the structure and improves image recognition performance [53]. On the other hand, compared against the TS fuzzy system, it has fewer rules and shorter running time, and it works better. Inspired by these merits, in our paper we attempt to leverage FBLS for single image deraining task.

III. The Proposed Solution

A. Overview

Fig. 2 plots the overall framework of our solution. It first uses the dehazing algorithm to deal with the rain in the distance based on the dark channel prior, so as to highlight the rain streaks near to the lens. Then, our solution separates the dehazed image into the base and detail layers by using the low and high pass filtering, respectively. Meanwhile, it converts the separated three-channel detail layer *from* the RGB color space *to* the YCbCr color space. Later, it puts the Y-channel image of the detail layer into the fuzzy broad learning system (FBLS) to obtain the derained Y-channel image (notice that, in the training phase the corresponding Y-channel ground truth is also put into the FBLS). After that, our solution combines the Cb and Cr channel images (separated before) with the derained Y channel image, so as to restore the derained detail layer of these three channels. Then, it obtains the preliminary derained image by fusing the derained detail layer with the base layer. Finally, it superimposes the details extracted from the dehazed image with some transparency on the preliminary derained image, obtaining the final result.

B. Training

We need some preprocessing operations on the images before we train our model. In what follows, we first cover

Fig. 3: Effect of dehazing processing: (a) the initial image before dehazing; (b) the image after dehazing.

the details of the preprocessing operations, and then discuss the training process in detail.

1) Preprocessing for training: The preprocessing contains two main operations: (i) processing the rain in the distance; and (ii) separating the image. Next, we address them, respectively.

► *Processing the rain in the distance.* It is easy to understand that the rain in the image can be classified into two categories: (i) rain streaks, which are close to the lens; and (ii) the rain in the distance (i.e., far way to the lens). Therefore, one can model the rainy image as follows:

$$I = B + R_C + R_D \tag{1}$$

where I denotes the original rainy image, B denotes the clean background layer, R_C denotes the rain close to the lens, and R_D denotes the rain in the distance.

To reduce the complexity of model learning in the later phase, and to increase the final quality of rain removal, it is needed to process R_D in the original input image. To this end, we use the dark channel priority dehazing algorithm [55] to process R_D. Generally, it is based on the following equation:

$$Fr(x) = \frac{F(x) - A}{\max(tr(x), T_0)} + A \tag{2}$$

where $Fr(x)$ is the dehazed image, $F(x)$ is the input image, $tr(x)$ is the transmittance, T_0 is a threshold in [0,1], and A is the light composition of the global atmosphere, which is obtained based on the pixel in the dark channel. After the dehazing operation, R_C becomes clearer, as shown in Fig. 3. This way, in the subsequent steps it would be easier to process R_C, and is helpful to improve the accuracy of the final resultant image.

► *Separating the image.* Given an image, one can separate it into two layers: (i) the base layer, which has the low frequency; and (ii) the detail layer, which has the high frequency. We can use the rolling guidance filter [56] to extract the high-pass detail layer and low-pass base layer of the rainy image.

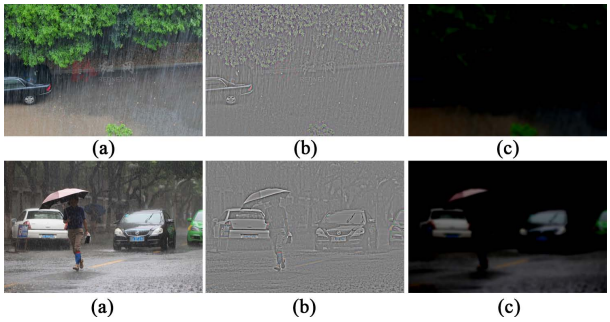


Fig. 4: Separating the rainy image: (a) preprocessed images to be separated; (b) the separated detail layer; (c) the separated base layer.

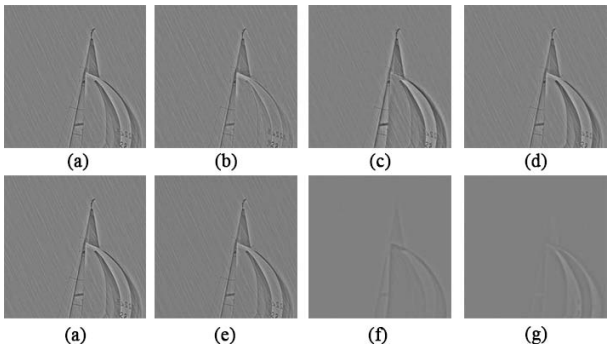


Fig. 5: Channel separation of RGB color space and YCbCr color space: (a) the detail layer of the un-separated channel; (b-d) channel R, G, and B in the RGB color space, respectively; (e-g) channel Y, Cb, and Cr in the YCbCr color space, respectively.

Compared with the smooth base layer, the detail layer looks more sharp and contains more rain streaks, as shown in Fig. 4. As the reader may know, image sparsity has been used in many computer vision applications. It is easy to understand that, when one uses sparse data for network training, it will accelerate network convergence speed, and significantly reduce network training time. To this end, we use the detail layer of sparse rainy image for network training, which simplifies the complexity of network training data. In addition, unlike channels of RGB color space in which all of them contain rain streaks, only Y channel of YCbCr color space contains rain streaks, as shown in Fig. 5. This implies that network training can be carried out on the Y channel of YCbCr color space.

2) Training via FBLS: After preprocessing, the information extracted from the Y channel in the YCbCr color space is taken as the training data, denoted as $X = \{x_1, x_2, ..., x_N\}^T \in \mathbb{R}^{n \times m}$. In the meanwhile, we also use the corresponding rainless Y-channel image in the YCbCr color space as the training ground truth. These paired training data is sent to the fuzzy broad learning system (FBLS). Fig. 6 illustrates the detailed structure of our training model. Generally, it consists of three layers: (i) input layer, which imports the training data X; (ii) data processing layer, which is mainly composed of fuzzy subsystems and enhanced node groups;

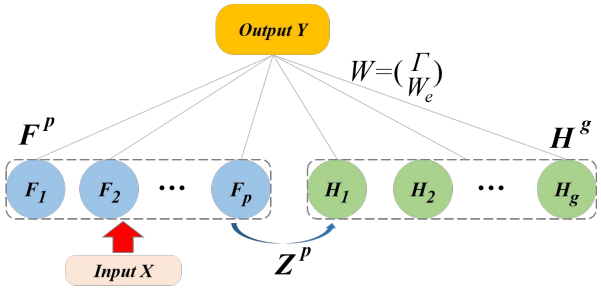


Fig. 6: The structure of our training model.

here the input of the enhancement node groups are transformed from the intermediate output of the fuzzy subsystems; and (iii) output layer, which utilizes the outputs of fuzzy subsystems and enhancement node groups to generate derained Y-channel image in the YCbCr color space. Note that, in the training phase the output layer is the ground truth image. Among these three layers, the data processing layer is much more complicated than other layers. More specifically, we need to obtain F^p, H^g and Z^p as shown in Fig. 6.

► *Obtaining* Z^p. In our method, we use first-order Takagi-Sugeno (TS) fuzzy model to map input $x_s = (x_{s1}, x_{s2}, ..., x_{sN})$ to the ith fuzzy subsystem, where $s = 1, 2, ..., p$. Each fuzzy subsystem has k^i fuzzy rules, as shown in Fig. 7. Formally, it is defined as

$$z^i_{sk} = f^i_k(x_{s1}, x_{s2}, ..., x_{sN}) = \sum_{t=1}^{N} \alpha^i_{kt} x_{st} \tag{3}$$

where α^i_{kt} refers to the coefficient, which is generated randomly in [0,1]; $k = 1, 2, ..., k^i$ denotes the fuzzy rule of the i-th fuzzy system.

Another element in the subsystem is the *weighted fire strength*, which is defined as:

$$\omega^i_{sk} = \frac{\prod_{t=1}^{n} \mu^i_{kt}(x_{st})}{\sum_{k=1}^{k^i} \prod_{t=1}^{n} \mu^i_{kt}(x_{st})} \tag{4}$$

where $\mu^i_{kt}(x_{st})$ refers to the Gaussian membership function, which is computed as:

$$\mu^i_{kt}(x_{st}) = e^{-(\frac{x_{st} - c^i_{kt}}{\sigma^i_{kt}})^2} \tag{5}$$

where the center c^i_{kt} can be obtained by using k-means clustering algorithm over the training set; and σ^i_{kt} is a harmonic parameter.

With the weighted fire strength ω^i_{sk} and fuzzy rule z^i_{sk}, we can derive the intermediate output value Z_{si} of the s-th training sample in the i-th fuzzy subsystem. That is,

$$Z_{si} = (w^i_{s1} z^i_{s1}, w^i_{s2} z^i_{s2}, ..., w_{sk^i} z^i_{sk^i}) \tag{6}$$

Then, for all training samples, the intermediate output of all training samples in the i-th fuzzy subsystem can be obtained as follows:

$$Z_i = (Z_{1i}, Z_{2i}, ..., Z_{pi}), i = 1, 2, ..., \tag{7}$$

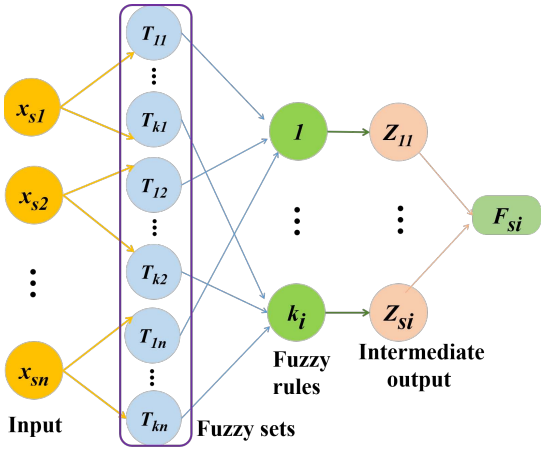


Fig. 7: Illustration of the TS fuzzy subsystem.

Finally, for all the subsystems, the intermediate output Z^p can be obtained as follows:

$$Z^p = (Z_1, Z_2, ..., Z_p) \in \mathbb{R}^{N \times (K_1 + K_2 + ... + K_p)} \tag{8}$$

► *Obtaining* H^g. The enhanced node's output H_v ($v = 1, 2, ..., g$) can be derived from the intermediate output Z^p. Specifically, it is defines as follows:

$$H_v = \xi_v(Z^p \omega_v + \beta_v) \in \mathbb{R}^{N \times L_v} \tag{9}$$

where ω_v is the weight of the enhanced node, and β_v is the bias term; the values of these two items are generated randomly in (0,1). In addition, $\xi(\cdot)$ refers to a nonlinear transformation function, and L_v denotes the number of neurons in group v of enhanced nodes.

Then, for the output H^g of all enhancement nodes, it can be computed as:

$$H^g = (H_1, H_2, ..., H_g) \in \mathbb{R}^{N \times (L_1 + L_2 + ... + L_g)} \tag{10}$$

► *Obtaining* F^p. As for the s-th training sample of the i-th fuzzy subsystem, its output is F_{si}, which is computed as:

$$\begin{aligned} F_{si} &= (\sum_{k=1}^{k^i} (\gamma^i_{k1} \omega^i_{sk} z^i_{sk}), ..., \sum_{k=1}^{k^i} \gamma^i_{kC} \omega^i_{sk} z^i_{sk}) \\ &= \sum_{t=1}^{n} \alpha^i_{kt} x_{st} (\omega^i_{s1}, \ldots, \omega^i_{sk^i}) \begin{pmatrix} \gamma^i_{11} & \cdots & \gamma^i_{1C} \\ \vdots & & \vdots \\ \gamma^i_{k^i 1} & \cdots & \gamma^i_{k^i C} \end{pmatrix} \end{aligned} \tag{11}$$

Then, for all n training samples of the i-th fuzzy subsystem, it is computed as:

$$F_i = (F_{1i}, F_{2i}, ..., F_{pi})^T = D \Omega^i \gamma^i \tag{12}$$

where

$$D = diag\left(\sum_{t=1}^{n} \alpha^i_{kt} x_{1t}, ..., \sum_{t=1}^{n} \alpha^i_{kt} x_{pt} \right),$$

Algorithm 1 The training process of FBLS

Input: Training samples (X, Y), p fuzzy subsystems , number of fuzzy rules k^i , enhancement nodes L_v, and g enhancement node groups.
Output: The weight W

Randomly initialize α_{kt}^i in [0,1] and set $\sigma_{kt}^i = 1$
while $i <= p$ **do**
  Calculate k^i centers
  Initialize Gaussian member function, and set $\sigma_{kt}^i = 1$
  for $s = 1; s \leq N$ **do**
    Calculate Z_{si} using Eq. 6
    Calculate F_{si} using Eq. 11
  end for
  Obtain Z_i using Eq. 7
  Calculate F_i using Eq. 12
  $i = i + 1$
end while
Obtain Z^p using Eq. 8
for $v = 1; v \leq g$ **do**
  Randomly initialize ω_v and β_v
  Calculate H_v
end for
Obtain H^g using $H^g = (H_1, H_2, ..., H_g)$;
Obtain F^p using Eq. 13
Calculate the pseudo-inverse $A^+ = (\lambda I + AA^T)^{-1}A^T$;
return $W = A^+O$;

$$\Omega^i = \begin{pmatrix} \omega_{11}^i & \cdots & \omega_{1k^i}^i \\ \vdots & & \vdots \\ \omega_{p1}^i & \cdots & \omega_{pk^i}^i \end{pmatrix},$$

$$\gamma^i = \begin{pmatrix} \gamma_{11}^i & \cdots & \gamma_{lC}^i \\ \vdots & & \vdots \\ \gamma_{k^i1}^i & \cdots & \gamma_{k^iC}^i \end{pmatrix}$$

Then, for all subsystems, the output F^p is computed as:

$$\begin{aligned} F^p &= \sum_{i=1}^{p} F_i = \sum_{i=1}^{p} D\Omega^i\gamma^i = D(\Omega^1, ..., \Omega^p) \begin{pmatrix} \gamma^1 \\ \vdots \\ \gamma^p \end{pmatrix} \\ &= D\Omega\Gamma \end{aligned} \tag{13}$$

where Γ denotes $\left((\gamma^1)^T, \ldots, (\gamma^p)^T\right)^T$, $\Omega = (\Omega^1, ..., \Omega^p)$.

► *Obtaining* W. On the basis of F^p and H^g, we can build the following equation:

$$\begin{aligned} O &= F^p + H^g W_e \\ &= D\Omega\Gamma + H^g W_e \\ &= (D\Omega|H^g)\begin{pmatrix} \Gamma \\ W_e \end{pmatrix} \\ &= (D\Omega|H^g)W \end{aligned} \tag{14}$$

where O denotes the output of the FBLS, and $W = \binom{\Gamma}{W_e}$. Note that, O in the training phase is the ground truth. Based on the above equation, we can obtain W by the pseudo-inverse. That is,

$$\begin{aligned} W &= (D\Omega|H^m)^+O \\ &= A^+O \end{aligned} \tag{15}$$

where A^+ is computed as

$$A^+ = (D\Omega|H^g)^+ \tag{16}$$

We also note that, the pseudo-inverse calculation is somewhat time-consuming, we use the following method [37] to optimize the calculation. That is,

$$A^+ = \lim_{\lambda \to 0}(\lambda I + AA^T)^{-1}A^T \tag{17}$$

where λ is an equilibrium factor. By combing Eqs. 15 and 17, we can get the weight W. The overall process of training on the FBLS is summarized in Algorithm 1.

C. Generating Rainless Map

As for the test phase, we also preprocess the input image as described earlier. That is, we also process R_D and separate the image into the (low pass) base layer and the (high pass) detail layer via a rolling guidance filter. Later, we put the Y-channel image of the detail layer into the fuzzy broad learning system (FBLS) to obtain the derained Y channel image, which is then combined with the Cb and Cr channel images, obtaining the derained detail layer. Then, we fuse the derained detail layer and the (low pass) base layer to obtain a preliminary derained image. Formally, it is described as follows:

$$I_{pre} = I_{deD} + I_b$$

where I_{pre}, I_{deD} and I_b denote the preliminary derained image, the derained detail layer, and the base layer, respectively. Fig. 8 shows an example of merging these two layers.

► *Further enhancement.* In the meanwhile, we employ the guided filter [57] to extract the details from the dahazed image[1]. For clearness, we denote the extracted detail part as I_{edp}.

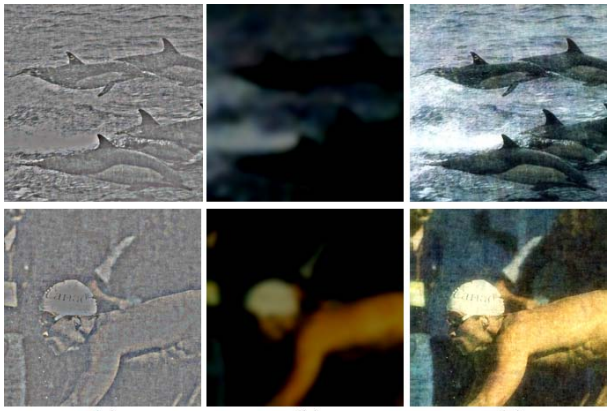


Fig. 8: Example of merging the base layer and the derained detail layer. (a) The detail layer after rain removal. (b) Rainless base layer. (c) preliminary derained image.

[1]The guided filter is simpler and quicker than the Rolling guidance filter [56]. Nevertheless, we here can also use the latter, since the visual effect obtained by these two filters are similar.

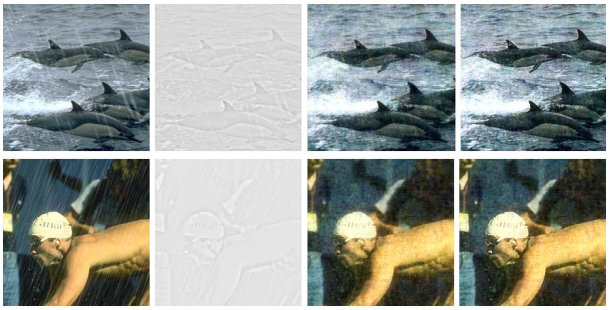


Fig. 9: Example of the enhancement processing. (a) Dehazed rainy image. (b) Seventy percent transparency image details extracted from dehazed rainy image. (c) Preliminary derained image. (d) Enhanced image, i.e., final result.

To further suppress the rainwater component and highlight other detailed components, we adjust the pixel transparency of I_{edp} based on the following equation:

$$I_{deT} = (I_{edp} \times Alpha + 127)/255 \tag{18}$$

where I_{deT} is the detail image after adjusting the transparency, and $Alpha$ refers to the transparency. Note that, the numbers 127 and 255 can be viewed as the color values.

Then, we superimpose the adjusted detail image I_{deT} as a blend color onto the preliminary derained image as the base color to further improve the visual effect. The superposition operation is as follows:

$$B = \begin{cases} (I_{pre} \times I_{deT})/255 & \text{if } I_{pre} \leq 128 \\ 255 - (I^*_{pre} \times I^*_{deT})/128 & \text{if } I_{pre} > 128 \end{cases} \tag{19}$$

where $I^*_{pre} = 255 - I_{pre}$, and $I^*_{deT} = 255 - I_{deT}$, B denotes the final image. Fig. 9 shows an example of the enhanced image.

IV. Experiments

In this section, we first introduce the experimental settings, and then cover the experimental results. Finally, we summarize our findings.

A. Experimental Setup

As the reader may know, in the field of single image deraining, there are no available large-scale real rainy image datasets with ground-truth images. In addition, it is clear that large-scale data can benefit to model training. To this end, we followed the prior work [58], [29], [30], [19], and also used the synthetic dataset for model training. The dataset used in our paper contains 20,800 images with 224×224 size. We divided the dataset into two parts: one part contains 16,640 images, which are used as the training set. The other part contains $4,160$ images, which are used as the test set. In addition, we also employed two other synthetic color images for test, since these two images are widely used in the prior work [26], [30]. For clearness, we call them *img1* and *img2*, respectively. Besides the synthetic test images, we also downloaded the real world rainy images from the site (https://image.baidu.com/) as the test images.

Following prior works [59], [60], we adopted two widely used evaluation metrics: (i) Peak Signal to Noise Ratio (PSNR), and (ii) Structural Similarity Index Measurement (SSIM). Generally, the higher the PSNR (SSIM) value is, the better a method is.

In the experiments, the equilibrium factor λ was set to 10^{-9}, and the harmonic parameter σ^i_{kt} was set to 1. We used the "sigmoid" function as the activation function of the enhancement node. We used the rectified linear unit as the nonlinear transformation function $\xi(\cdot)$. We set the transparency $Alpha$ to 70%, according to extensive offline tests.

As for the number of fuzzy subsystems p, the number of enhancement nodes g, and the number of fuzzy rules k^i, we initially set them to 5, 200, and 3 empirically. After that, we gradually increase them to 150, 1300, and 30, respectively. We found that the average SSIM and PSNR values over the test set (i.e., 4,160 images) increase when we increase the values of these parameters, and these values tend to be stable when p, g, and k^i are increased to 125, 1100, and 23. Specifically, the average SSIM and PSNR values over the test set are 0.8531 and 24.01, respectively. In view of this, in the remaining experiments, we set p, g, and k^i to 125, 1100, and 23 respectively, unless stated otherwise.

We compared our proposed solution with both traditional rain removal methods and also deep learning based ones. For some algorithms (e.g., Auto-SP [26], CCRR [5], JBLO [4]) we used the implementations provided by authors. As for algorithms without publicly available implementations, we directly compared the results presented in their papers. The codes of our solution were written using Matlab 2016Ra. Our experiments were conducted on a computer with 4-core 2.80Ghz CPU and 64GB RAM.

B. Experimental Results

Synthetic rainly images with ground-truth. Table I presents the quantitative comparison of our solution and state-of-the-art algorithms over *img1*. From this table, it can be seen that, our solution outperforms most of these state-of-the-art algorithms (e.g., Low-rank [25], JORDER [61], CNN [29], GMMLP [19], Auto-SP [26], Dis-SP [18]), demonstrating the feasibility and competitiveness of our solution. In the meanwhile, we observed that our solution is slightly inferior to several algorithms (e.g., CCRR [5], JBLO [4]), in terms of SSIM and PSNR. A possible reason is that our method does not leverage new rain streak priors. Another possible reason is that the fuzzy broad learning system is still a relatively young system, appealing to much more improvements. Nevertheless, we have to mention that the visual effect of our solution is highly close to them (i.e., CCRR and JBLO), since the derained result of our solution preserves well the details (e.g., wrinkles on the clothes) while removing rain streaks effectively, as shown in Fig. 10. Furthermore, we observed that the experimental results over *img2* exhibit the similar performance, as shown in Table II and Fig. 11. This further validates the feasibility and competitiveness of our solution.

TABLE I: Quantitative comparison on *img1*. The top three results have been highlighted in red, blue and green.

Method	Low-rank	DDN	JORDER	CNN	GMMLP	Auto-SP	Dis-SP	CCRR	JBLO	Ours
SSIM	0.8153	0.7906	0.6881	0.7584	0.7638	0.7853	0.8419	0.8662	0.8789	0.8583
PSNR	22.74	25.78	22.8	20.00	23.27	21.23	24.84	25.56	29.18	25.18

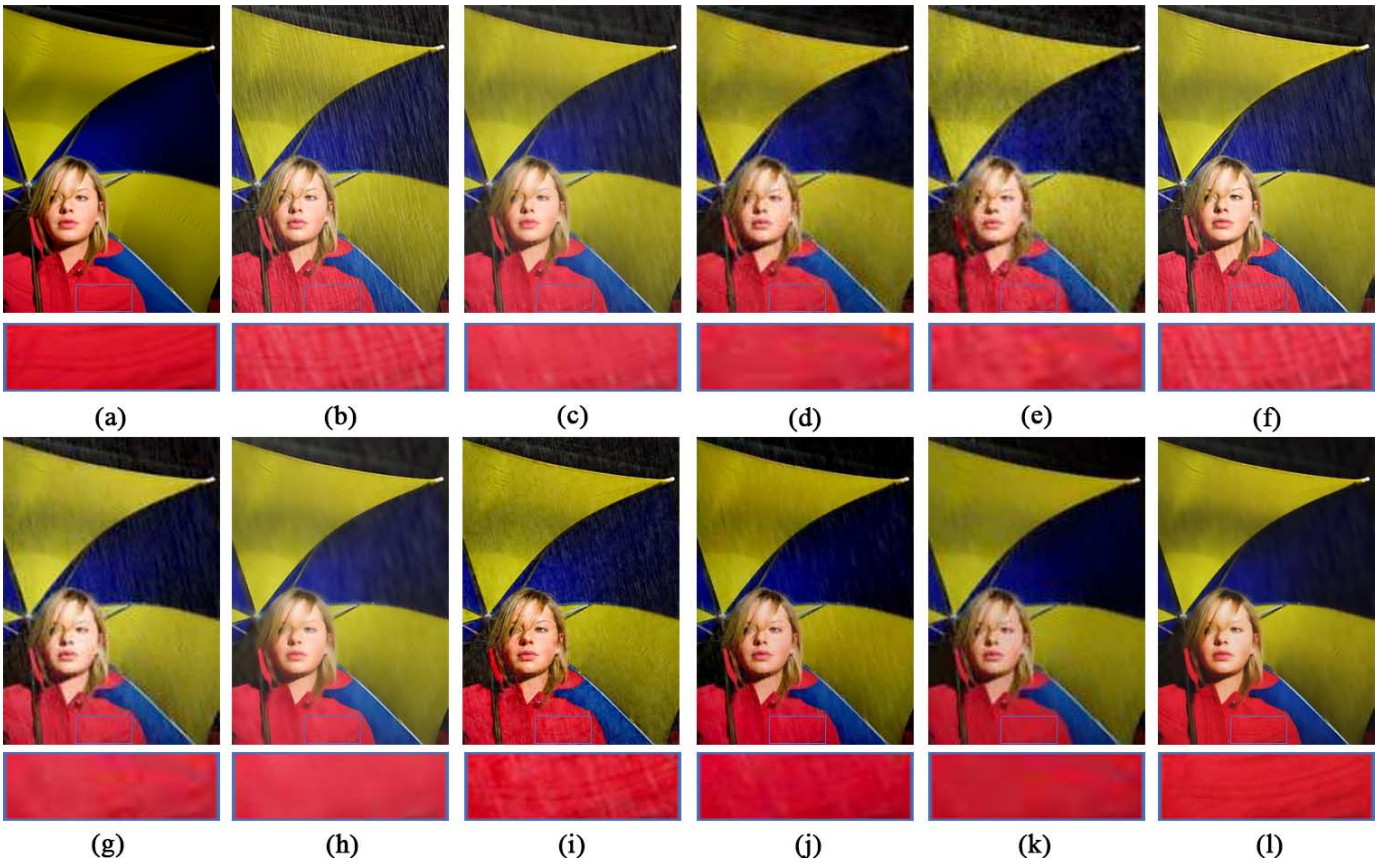


Fig. 10: Qualitative comparison over *img1*. The bottom part of each subfigure is the partial enlarged effect of the corresponding image. (a) Rainless ground truth; (b) input rainy image; (c) Low-rank [25]; (d) DDN [12]; (e) JORDER [61]; (f) CNN [29]; (g) GMMLP [19]; (h) Auto-SP [26]; (i) Dis-SP [18]; (j) CCRR [5]; (k) JBLO [4]; (l) our method.

Real world rainy images. Fig. 12 presents the visual comparison with representative algorithms on several real world rainy images. It can be seen that, (i) some methods (e.g., Dis-SP [18]) preserve most of the details, while the rain streaks are not removed sufficiently. (ii) Some methods (e.g., GMMLP [19] and SL [62]) make the derained image too smooth, losing many details. (iii) Some methods (e.g. SL [62]) make the color of the derained image too dark. In contrast, our method efficiently removes the rain streaks while preserving the details of the images well. This essentially demonstrates that our method is also effective for processing real world rainy images. On the other hand, this implies that our method has good generalization performance (since we here tested on the real world rainy image while we trained our model using synthetic rainy images). Note that, we did not conduct qualitative comparison on real world rainy images, since the ground truth images are not available.

Runing time. Table III summarizes the CPU running time of our method with state-of-the-art methods including GMMLP [19], SL [62], Dis-SP [18], CNN [29], JORDER [61], DNN [12], ID_CGAN [63], DDC [58]. It can be seen that, among these algorithms our method takes the least running time for different image sizes. Generally, it is 2$\times$ faster than the strongest competitor (i.e., ID_CGAN [63]). This demonstrates that our method has a very competitive execution efficiency. Note that, the execution efficiency is another important metric

TABLE II: Quantitative comparison on *img2*. The top three results have been highlighted in red, blue and green.

Method	Low-rank	DDN	JORDER	CNN	GMMLP	Auto-SP	Dis-SP	CCRR	JBLO	Ours
SSIM	0.8215	0.8071	0.6424	0.7326	0.8138	0.8181	0.7915	0.8526	0.8610	0.8457
PSNR	21.87	28.30	21.40	21.29	25.43	21.74	24.15	26.05	29.16	25.47

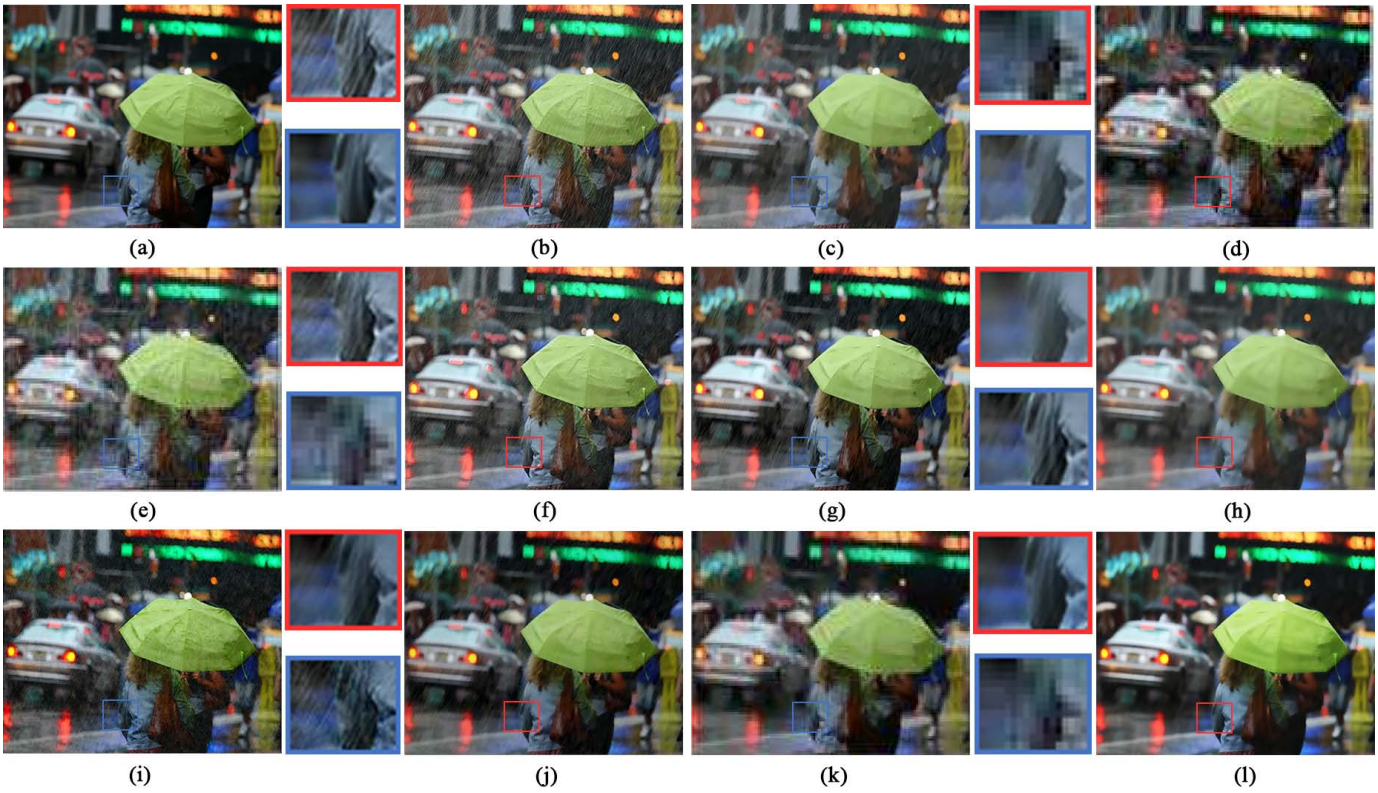


Fig. 11: Qualitative comparison over *img2*. The bottom part of each subfigure is the partial enlarged effect of the corresponding image. (a) Rainless ground truth; (b) input rainy image; (c) Low-rank; (d) DDN; (e) JORDER; (f) CNN; (g) GMMLP; (h) Auto-SP; (i) Dis-SP; (j) CCRR; (k) JBLO; (l) our method.

TABLE III: Running time of different methods (seconds).

Image Size	GMMLP	SL	Dis-SP	CNN	JORDER	DDN	DDC	ID_CGAN	Ours
250×250	196	68	53	1.32	48	1.81	0.98	0.15	**0.073**
500×500	942	76	230	2.86	88	13.27	4.04	0.55	**0.29**
750×750	1374	99	782	5.47	149	39	9.12	1.17	**0.57**

to evaluate an algorithm, since it is very vital for many real-time applications.

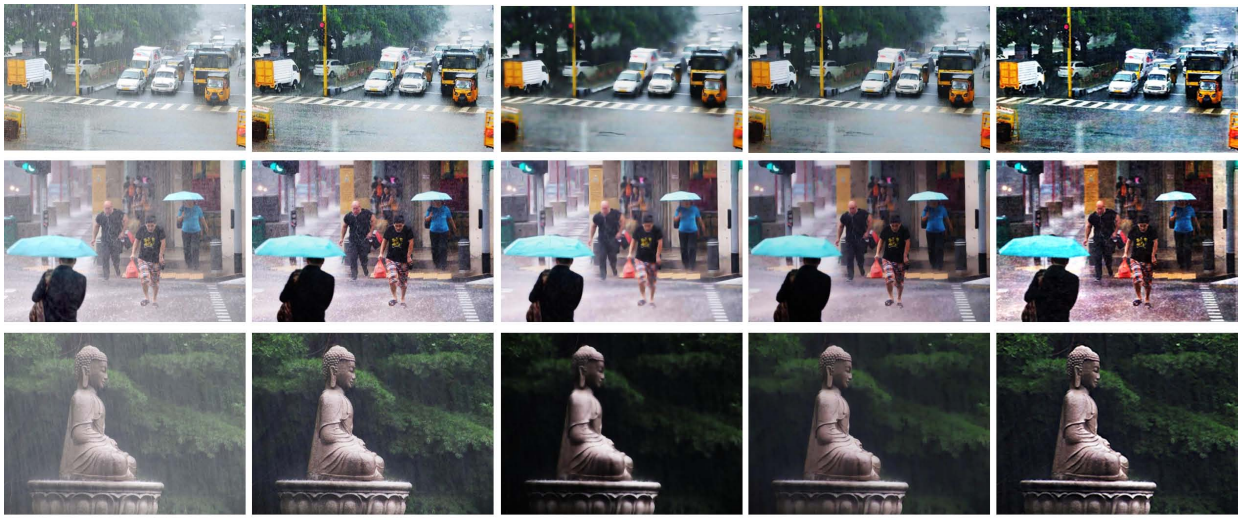
C. Further Comparison

To further examine the performance of our proposed method, Table IV below covers the comparison results on the test dataset with 4,160 images (recall Section IV-A), and on Rain100H and Rain100L datasets [61], respectively. Generally, our proposed method outperforms most of the algorithms. This finding is basically consist with the results presented in Section IV-B.

Besides, we also conduct a user study based on images chosen from these datasets. Generally speaking, we randomly extract 20 images from the above datasets, and then execute algorithms to perform deraining operation on these selected images, respectively; and finally we invite 10 persons to evaluate the quality of the generated de-rained images. The invited persons are allowed to give scores (0∼10) for each generated image. For ease of comparison, we compute and report the average score. The results are shown in Table V. From this table, we can see that the evaluation score of our solution is higher than most of algorithms, and it is very near to the scores of JORDER and DDN. Note that, compared to the competitors, our solution does not rely on the rain streak priors, and it can automatically detect and remove rain streaks. This can be also viewed as an advantage of our solution.

D. Discussion and Summary

Recall Section IV-A, we trained our model and obtained the average SSIM and PSNR values over large-scale test images with ground truth. It is meaningful to examine the visual effects of such test images. To this end, we show several representative rainy images and the derained results, as shown in Fig. 13. It can be seen that, these rainy images (cf., the 1st column) have different rain intensity (e.g., heavy rain, light rain), and the rain streaks are in various directions (e.g., from right-top to left-bottom, from left-top to right-bottom). Nevertheless, our method exhibits good deraining performance on these challenging images, since it removes effectively the rain streaks and preserves most of details in the original



(a) (b) (c) (d) (e)

Fig. 12: Comparison of different rain removal methods on real world rainy images. (a) Real world rainy image; (b) derained result using Dis-SP [18]; (c) derained result using SL [62]; (d) derained result using GMMLP [19]; (e) derained result using our method.

TABLE IV: Quantitative comparison on the testing set with 4,160 images, and on the Rain100H and Rain100L datasets. The top three results have been highlighted in red, blue and green.

Method	Auto-SP	GMMLP	CNN	Dis-SP	DDN	JORDER	Ours
4160-SSIM	0.62	0.75	0.63	0.72	0.87	0.90	0.85
4160-PSRN	20.1	23.32	23.05	20.9	30.36	29.8	24.01
Rain100H-SSIM	0.40	0.43	0.37	0.54	0.76	0.80	0.74
Rain100H-PSNR	13.78	15.05	13.21	17.55	21.92	25.21	20.13
Rain100L-SSIM	0.70	0.86	0.81	0.86	0.93	0.97	0.89
Rain100L-PSNR	23.13	28.65	23.70	23.39	29.39	35.23	27.36

TABLE V: User Study with 20 randomly selected images. The score refers to the average score evaluated by 10 persons.

Method	Auto-SP	GMMLP	CNN	Dis-SP	DDN	JORDER	Ours
Evaluation Score	7.14	7.52	7.43	7.15	8.22	8.17	8.15

input images. Note that, it is possible that some generated images look over-enhanced. In this case, one can improve it by adjusting the transparency of the detailed image.

Another thing related to training is the training efficiency. As the reader may know, many existing deraining methods usually take a long time to train their models (e.g., CNN [29] consumes approximately two days on a PC with Intel Core i5 CPU 4460, 8GB RAM and NVIDIA Geforce GTX 750, DDN [30] consumes over 8 hours to train their model, DNN [12] consumes approximately 14 hours to train their model). In contrast, training our model used about half an hour. This can be viewed as another advantage of our method.

Summary. We find that, (i) our solution can achieve singe image rain removal efficiently and it outperforms several state-of-the-art algorithms (e.g., Low-rank [25], JORDER [61], CNN [29], GMMLP [19], Auto-SP [26], Dis-SP [18]), in terms of PSNR and SSIM. (ii) Although our solution is slightly inferior to several strong competitors (e.g., CCRR [5], JBLO [4]) in terms of PSNR and SSIM, the visual effect of our solution is close to the ones generated by these strong competitors. (iii) Our solution has the good generality performance. Specifically, we trained our model using the synthetic dataset, while our solution can exhibit good deraining performance on real world test images. (iv) Compared against existing algorithms, our solution has a higher efficiency in terms of training time and running time. Specifically, it takes about 0.5 hours to train our model, and on average our solution can perform singe image rain removal in 0.07, 0.29, and 0.57 seconds for image sizes with 250×250, 500×500, and 750×750, respectively.

V. Conclusion

In this paper we have presented a single image deraining method. Our solution first pre-processes the input image based on the dark channel prior dehazing algorithm, so as to highlight the rain streaks hidden in the in the distance (or "fog"). After that, it separates the dehazed image into detail layer and

Fig. 13: Rainy images with different cases such as rain in different directions, heavy rain and light rain scenarios. (a) Preprocessed image (b) Rainless ground truth. (c) The derained image using our method.

base layer. Particularly, it uses the fuzzy broad learning to process the high-frequency detail layer of the rainy image. Then, the processed detail layer is combined with the base layer to obtain a preliminary derained image. Finally, it superimposes the details extracted from the dehazed image with a certain transparency on the preliminary derained image, obtaining the final result. We have conducted empirical study based on both real and synthetic rainy images. Experimental results show that our solution can generate high-quality derained images, outperforming several state-of-the-art algorithms. Moreover, our solution has a higher efficiency in terms of training time and running time. A challenging problem is to jointly train the dehazing and deraining based on the fuzzy broad learning system, we leave this opening problem as the future work.

ACKNOWLEDGMENT

We thank the editors and anonymous reviewers very much for their efforts in evaluating our manuscript. This work was partially supported by the NSFC (No. U1811264, 61972425, 61972157, 61775139, 61872242), the National Key R&D Program of China (No. 2018YFB1004400), the Shanghai Science and Technology Innovation Action Plan Project (No. 16111107502, 17511107203), the Research Grants Council of Hong Kong (No. 28200215), the Key R&D Program of Guangdong Province (No. 2018B010107005, 2019B010120001), the National High-tech R&D Program of China (863 Program) (No. 2015AA015904), the Key Program for International S&T Cooperation Project of China (No. 2016YFE0129500), the Science and Technology Commission of Shanghai Municipality (No. 16DZ0501100, 17411952600).

REFERENCES

[1] Y. Song, J. Li, X. Wang, and X. Chen, "Single image dehazing using ranking convolutional neural network," *IEEE Trans. Multimedia*, vol. 20, no. 6, pp. 1548–1560, 2018.
[2] L. Shi, B. Chen, S. Huang, A. O. Larin, O. Seredin, A. V. Kopylov, and S. Kuo, "Removing haze particles from single image via exponential inference with support vector data description," *IEEE Trans. Multimedia*, vol. 20, no. 9, pp. 2503–2512, 2018.
[3] Z. Wang, L. Ma, X. Lin, and H. Zhong, "Saliency detection via multi-center convex hull prior," in *ICASSP*, 2018, pp. 1867–1871.
[4] L. Zhu, C. Fu, D. Lischinski, and P. Heng, "Joint bi-layer optimization for single-image rain streak removal," in *IEEE International Conference on Computer Vision*, 2017, pp. 2545–2553.
[5] H. Zhang and V. M. Patel, "Convolutional sparse and low-rank coding-based image decomposition," *IEEE Trans. Image Processing*, vol. 27, no. 5, pp. 2121–2133, 2018.
[6] Z. Ling, J. Gong, G. Fan, and X. Lu, "Optimal transmission estimation via fog density perception for efficient single image defogging," *IEEE Trans. Multimedia*, vol. 20, no. 7, pp. 1699–1711, 2018.
[7] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu, "Semi-supervised transfer learning for image rain removal," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019, pp. 3877–3886.
[8] X. Lin, Z. Wang, L. Ma, and X. Wu, "Saliency detection via multi-scale global cues," *IEEE Trans. Multimedia*, vol. 21, no. 7, pp. 1646–1659, 2019.
[9] Z. Gao, J. Xue, W. Zhou, S. Pang, and Q. Tian, "Democratic diffusion aggregation for image retrieval," *IEEE Trans. Multimedia*, vol. 18, no. 8, pp. 1661–1674, 2016.
[10] S. Wang, K. Gu, S. Ma, W. Lin, X. Liu, and W. Gao, "Guided image contrast enhancement based on retrieved images in cloud," *IEEE Trans. Multimedia*, vol. 18, no. 2, pp. 219–232, 2016.
[11] H. Xu, G. Zhai, X. Wu, and X. Yang, "Generalized equalization model for image enhancement," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 68–82, 2014.
[12] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. W. Paisley, "Removing rain from single images via a deep detail network," in *CVPR*, 2017, pp. 1715–1723.
[13] B. Chen, S. Huang, and S. Kuo, "Error-optimized sparse representation for single image rain removal," *IEEE Trans. Industrial Electronics*, vol. 64, no. 8, pp. 6573–6581, 2017.
[14] J. Chen, C. Tan, J. Hou, L. Chau, and H. Li, "Robust video content alignment and compensation for rain removal in a cnn framework," in *CVPR*, 2018, pp. 6286–6295.
[15] T. Jiang, T. Huang, X. Zhao, L. Deng, and Y. Wang, "Fastderain: A novel video rain streak removal method using directional gradient priors," *IEEE Trans. Image Processing*, vol. 28, no. 4, pp. 2089–2102, 2018.
[16] S. Gu, D. Meng, W. Zuo, and L. Zhang, "Joint convolutional analysis and synthesis sparse representation for single image layer separation," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 1717–1725.
[17] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, pp. 695–704.
[18] L. Yu, X. Yong, and J. Hui, "Removing rain from a single image via discriminative sparse coding," in *ICCV*, 2015, pp. 3397–3405.
[19] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," in *CVPR*, 2016, pp. 2736–2744.
[20] X. Ding, L. Chen, X. Zheng, Y. Huang, and D. Zeng, "Single image rain and snow removal via guided l0 smoothing filter," *Multimedia Tools and Applications*, vol. 75, no. 5, pp. 2697–2712, 2016.
[21] C. Wang, M. Shen, and Y. Chen, "Rain streak removal by multi-frame-based anisotropic filtering," *Multimedia Tools and Applications*, vol. 76, no. 2, pp. 2019–2038, 2016.
[22] 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 Processing*, vol. 26, no. 8, pp. 3936–3950, 2017.
[23] J. Kim, C. Lee, J. Sim, and C. Kim, "Single-image deraining using an adaptive nonlocal means filter," in *ICIP*, 2013, pp. 914–917.
[24] J. Xu, W. Zhao, P. Liu, and X. Tang, "An improved guidance image based method to remove rain and snow in a single image," *Computer and Information Science*, vol. 5, no. 3, pp. 49–55, 2012.
[25] Y. Chen and C. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *ICCV*, 2013, pp. 1968–1975.

[26] L. W. Kang, C. W. Lin, and Y. H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1742–1755, 2012.

[27] Y. Fu, L. Kang, C. Lin, and C. Hsu, "Single-frame-based rain removal via image decomposition," in *ICASSP*, 2011, pp. 1453–1456.

[28] D. Chen, C. Chen, and L. Kang, "Visual depth guided color image rain streaks removal using sparse coding," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 24, no. 8, pp. 1430–1455, 2014.

[29] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2944–2956, 2016.

[30] S. Liang, Z. Yue, C. Quan, F. Fan, and M. Jie, "Deep joint rain and haze removal from single images," in *ICPR*, 2018, pp. 2821–2826.

[31] Z. Fan, H. Wu, X. Fu, Y. Huang, and X. Ding, "Residual-guide network for single image deraining," in *ACM MM*, 2018, pp. 1751–1759.

[32] T. Zhao, B. Zhang, M. He, W. Zhang, N. Zhou, J. Yu, and J. Fan, "Embedding visual hierarchy with deep networks for large-scale visual recognition," *IEEE Trans. Image Processing*, vol. 27, no. 10, pp. 4740–4755, 2018.

[33] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters - improve semantic segmentation by global convolutional network," in *CVPR*, 2017, pp. 1743–1751.

[34] Z. Quan, Z. Wang, Y. Le, B. Yao, K. Li, and J. Yin, "An efficient framework for sentence similarity modeling," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 27, no. 4, pp. 853–865, 2019.

[35] Y. Le, Z. Wang, Z. Quan, J. He, and B. Yao, "Acv-tree: A new method for sentence similarity modeling," in *IJCAI*, 2018, pp. 4137–4143.

[36] Z. Quan, Y. Guo, X. Lin, Z. Wang, and X. Zeng, "Graphcpi: Graph neural representation learning for compound-protein interaction," in *BIBM*, 2019, pp. 717–722.

[37] F. Shuang and C. L. P. Chen, "Fuzzy broad learning system: A novel neuro-fuzzy model for regression and classification," *IEEE Trans. Cybernetics*, vol. PP, no. 99, pp. 1–11, 2018.

[38] H. Wu and S. Feng, "Mixed fuzzy/boundary control design for nonlinear coupled systems of ODE and boundary-disturbed uncertain beam," *IEEE Trans. Fuzzy Systems*, vol. 26, no. 6, pp. 3379–3390, 2018.

[39] K. He, S. Jian, and X. Tang, "Single image haze removal using dark channel prior," in *CVPR*, 2009, pp. 1956–1963.

[40] P. C. Barnum, S. Narasimhan, and T. Kanade, "Analysis of rain and snow in frequency space," *International Journal of Computer Vision*, vol. 86, no. 2, pp. 256–274, 2010.

[41] J. Bossu, N. Hautière, and J. Tarel, "Rain or snow detection in image sequences through use of a histogram of orientation of streaks," *International Journal of Computer Vision*, vol. 93, no. 3, pp. 348–367, 2011.

[42] C. Son and X. Zhang, "Rain removal via shrinkage of sparse codes and learned rain dictionary," in *ICME Workshops*, 2016, pp. 1–6.

[43] W. Yang, R. T. Tan, J. Feng, J. Liu, S. Yan, and Z. Guo, "Joint rain detection and removal from a single image with contextualized deep networks," *IEEE Trans. Pattern Anal. Mach. Intell.*

[44] W. Liu, Z. Wang, B. Yao, M. Nie, J. Wang, R. Mao, and J. Yin, "Geographical relevance model for long tail point-of-interest recommendation," in *DASFAA*, 2018, pp. 67–82.

[45] W. Liu, Z. Wang, B. Yao, and J. Yin, "Geo-alm: POI recommendation by fusing geographical information and adversarial learning mechanism," in *IJCAI*, 2019, pp. 1807–1813.

[46] Y. Gui, Y. Tian, D. Zeng, Z. Xie, and Y. Cai, "Reliable and dynamic appearance modeling and label consistency enforcing for fast and coherent video object segmentation with the bilateral grid," *IEEE Trans. Circuits Syst. Video Techn.*

[47] X. Lin, Z. Wang, X. Tan, M. Fang, N. N. Xiong, and L. Ma, "MCCH: A novel convex hull prior based solution for saliency detection," *Inf. Sci.*, vol. 485, pp. 521–539, 2019.

[48] Z. Wang, L. Ma, X. Lin, and X. Wu, "MSGC: A new bottom-up model for salient object detection," in *ICME*, 2018, pp. 1–6.

[49] L. Yu, J. Yu, and Q. Ling, "Bltrcnn-based 3-d articulatory movement prediction: Learning articulatory synchronicity from both text and audio inputs," *IEEE Trans. Multimedia*, vol. 21, no. 7, pp. 1621–1632, 2019.

[50] Z. Wang, D. Wang, B. Yao, and M. Guo, "Probabilistic range query over uncertain moving objects in constrained two-dimensional space," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 3, pp. 866–879, 2015.

[51] Z. Wang, B. Yao, R. Cheng, X. Gao, L. Zou, H. Guan, and M. Guo, "Sme: explicit & implicit constrained-space probabilistic threshold range queries for moving objects," *GeoInformatica*, vol. 20, no. 1, pp. 19–58, 2016.

[52] M. Nie, Z. Wang, J. Yin, and B. Yao, "Reachable region query and its applications," *Inf. Sci.*, vol. 476, pp. 95–105, 2019.

[53] K. Dragan and L. Emil, "Identification of complex systems based on neural and takagi-sugeno fuzzy model," *IEEE Trans. Syst Man Cybern B Cybern*, vol. 34, no. 1, pp. 272–282, 2004.

[54] C. Chen and Z. L. , "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 1, pp. 10–24, 2018.

[55] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2341–2353, 2011.

[56] Z. Qi, X. Shen, X. Li, and J. Jia, "Rolling guidance filter," in *ECCV*, 2014, pp. 815–830.

[57] H. Kaiming, S. Jian, and T. Xiaoou, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, 2013.

[58] S. Li, W. Ren, J. Zhang, J. Yu, and X. Guo, "Fast single image rain removal via a deep decomposition-composition network," *CoRR*, vol. abs/1804.02688, 2018.

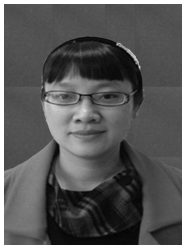
[59] B. Ouali, "Peak signal-to-noise ratio," *Electronics Letters*, vol. 44, pp. 800–801, 2008.

[60] W. Zhou, B. Alan-Conrad, S. Hamid-Rahim, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process*, vol. 13, no. 4, pp. 600–612, 2004.

[61] 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 *CVPR*, 2017, pp. 1685–1694.

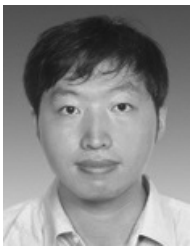
[62] De-An, H. L. Kang, Y. C. F. Wang, and C. Lin, "Self-learning based image decomposition with applications to single image denoising," *IEEE Transactions on Multimedia*, vol. 16, no. 1, pp. 83–93, 2013.

[63] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *CoRR*, vol. abs/1701.05957, 2017.



Xiao Lin received the Ph.D. degree in computer science from the Shanghai Jiao Tong University, Shanghai, China. She is currently an Associate Professor with the Department of Computer Science, Shanghai Normal University, Shanghai, China. She is also a Visiting Scholar with the Shanghai Jiao Tong University, Shanghai, China. She has authored and co-authored a set of research papers in internal journals and conferences such as IEEE TMM, Elsevier CAD, IEEE ICME, etc. Her current research interests include machine learning, computer vision, computer graphics, multimedia and image processing, etc.

Lizhuang Ma received the BS and PhD degrees from the Zhejiang University, Hangzhou, China, in 1985 and 1991, respectively. Currently, he is a distinguished professor and the head of the Digital Media and Data Reconstruction Laboratory with the Department of Computer Science and Engineering, Shanghai Jiao Tong University (SJTU), Shanghai, China. Before Joining SJTU, he worked as a full processor at the State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou, China. Meanwhile, he worked as a visiting processor in several organizations/institutions including Nayang Technological University at Singapore, Fraunhofer IGD at Germany. His research interests include digital media technology, computer graphics, image & video processing, etc.



Bin Sheng received the B.A. degree in English and the B.Eng. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, and the M.Sc. degree in software engineering from the University of Macau, Taipa, Macau, and the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Shatin, Hong Kong. He is currently an Associate Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include computer vision, multimedia and image processing, virtual reality, and computer graphics.

Zhi-Jie Wang received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, and did a posdoc at The Hong Kong Polytechnic University, Kowloon, Hong Kong. He is currently an Associate Professor at the College of Computer Science, Chongqing University (CQU), Chongqing, China. Before joining CQU, he worked at the Sun Yat-Sen University (SYSU), Guangzhou, China. His current research interests include data processing and analysis for spatial/temporal data, image/video data, etc. He has published over 30 research papers in venues such as IEEE TKDE, IEEE TMM, IEEE TPDS, IEEE/ACM TALSP, IJCAI, AAAI, ECAI, ICME, etc. He is a member of the CCF, IEEE, and ACM.

Wansheng Chen received B.Eng. degree in computer science from Xuzhou University of Technology, Xuzhou, China. He is currently pursuing the M.Eng. degree in computer science with the Department of Computer Science and Engineering, Shanghai Normal University, Shanghai, China. He is recently working as an intern at the Shanghai Rural Commercial Bank, Shanghai, China, where he is engaged in machine learning and big data processing. His research interests include machine learning, computer vision, and big data.