Single Image Dehazing Using Ranking Convolutional Neural Network

Yafei Song, Jia Li, *Senior Member, IEEE*, Xiaogang Wang, and Xiaowu Chen, *Senior Member, IEEE*

***Abstract*—Single image dehazing, which aims to recover the clear image solely from an input hazy or foggy image, is a challenging ill-posed problem. Analyzing existing approaches, the common key step is to estimate the haze density of each pixel. To this end, various approaches often *heuristically designed* haze-relevant features. Several recent works also automatically learn the features via directly exploiting convolutional neural networks (CNN). However, it may be insufficient to fully capture the intrinsic attributes of hazy images. To obtain effective features for single image dehazing, this paper presents a novel ranking convolutional neural network (Ranking-CNN). In Ranking-CNN, a novel ranking layer is proposed to extend the structure of CNN so that the statistical and structural attributes of hazy images can be simultaneously captured. By training Ranking-CNN in a well-designed manner, powerful haze-relevant features can be *automatically learned* from massive hazy image patches. Based on these features, haze can be effectively removed by using a haze density prediction model trained through the random forest regression. Experimental results show that our approach outperforms several previous dehazing approaches on synthetic and real-world benchmark images. Comprehensive analyses are also conducted to interpret the proposed Ranking-CNN from both the theoretical and experimental aspects.**

***Index Terms*—Single image dehazing, haze-relevant features, convolutional neural network, ranking layer.**

I. Introduction

IN REAL-WORLD scenarios, small particles suspending in the atmosphere (e.g., droplets and dusts) often scatter the light. As a consequence, the clarity of an image would be seriously degraded, which may decrease the performance of many multi-media processing systems, e.g., content-based image retrieval [11]. Image enhancement methods [48], [54] can only alleviate this problem slightly. It is still helpful to develop effective dehazing methods to recover the clear image from an input hazy or foggy image.

In the past decades, the problem of haze formation has been extensively studied in atmospheric optics [47]. It is widely acknowledged that a hazy image can be regarded as a convex combination of scene radiance and atmospheric light [2], [4], [10], [14], [32], [45], [49]. The combination coefficient is often called the *transmission*. As a result, the task of image dehazing can be formulated as *recovering the scene radiance from a hazy image by estimating the atmospheric light and the transmission*.

Under this formulation, two kinds of dehazing approaches have been proposed in the literature. Some of them propose to dehaze an image under the assistance of additional information, e.g., scene depth [19], images taken under different weathers [29], [30]. However, such additional information may not be always available, which prevents the further usage of these dehazing approaches in many real-world scenarios. On the contrary, some approaches propose to directly dehaze a single image, which is an ill-posed problem since the atmospheric light and the transmission need to be simultaneously recovered for each image pixel. To address this issue, these approaches often assume that the atmospheric light is constant for every pixel in one input image, so that it can be estimated first in a pre-processing step. After that, the dehazing process can be simplified as a transmission estimation problem. For instance, He *et al.* [14] propose the dark channel prior which is proved to be effective in transmission estimation. Tang *et al.* [45] incorporate four types of features to train a regression model for transmission prediction. Fattal [10] utilizes local color-lines prior in clear images to estimate the transmission. Berman *et al.* [2] further propose non-local haze-line prior. In many cases, these approaches achieve impressive performance. However, for each prior, there are often images which may not meet it. Therefore, the heuristic designed priors (or features) may be insufficient to fully capture the intrinsic attributes of hazy images.

Inspired by the impressive success of Convolutional Neural Networks (CNN) [22], e.g., image classification/annotation [21], [52], object detection [7], semantic segmentation [8], and image denoising [1], [53], this paper prefers to automatically learn the haze-relevant features from massive hazy images. Two recent works [4], [32] also hold the same basic idea and adopt CNN to perform image dehazing. Ren *et al.* [32] directly estimate the whole transmission map from an input image under the multi-scale FCN (fully convolutional networks) framework [24]. Cai *et al.* [4] use a regression network to estimate the

Manuscript received March 1, 2017; revised June 14, 2017 and September 7, 2017; accepted October 18, 2017. Date of publication November 8, 2017; date of current version May 15, 2018. This work was supported by the National Natural Science Foundation of China under Grants 61532003, 61672072, 61325011, and 61421003. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Leonel Sousa. *(Corresponding author: Xiaowu Chen.)*

Y. Song, X. Wang, and X. Chen are with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: songyf@buaa.edu.cn; wangxiaogang@buaa.edu.cn; chen@buaa.edu.cn).

J. Li is with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China, and also with the International Research Institute for Multidisciplinary Science, Beihang University, Beijing 100191, China (e-mail: jiali@buaa.edu.cn).

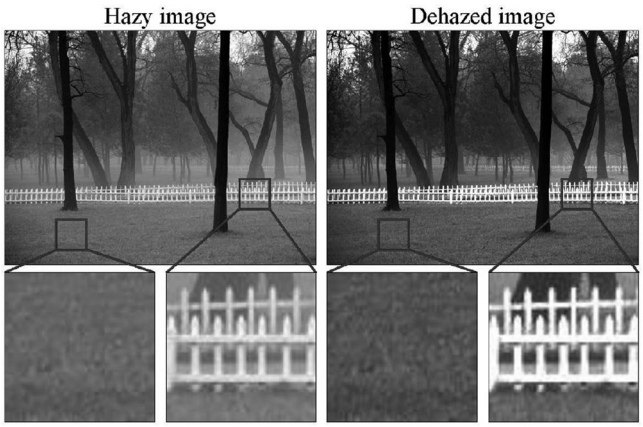


Fig. 1. Both statistical and structural attributes of image patches are useful for dehazing. For example, the grass patches can be dehazed according to their color statistics (*i.e.*, the statistical attributes of image patches), while the haze over fence can be removed according to their gradients (*i.e.*, the structural attributes of image patches).

transmission of each pixel from its local surrounding patch. However, these two works mainly exploit existing layers to construct their CNNs. In contrast, we propose a new layer, named ranking layer, derived from our insight on this problem, which can facilitate the learning process of haze-relevant features.

By analysing the mechanism of existing image dehazing methods, we find that statistical attributes are essential, e.g., dark channel prior [14], haze-line prior [2] and color-lines prior [10]. But the classical CNN, while capturing the structural attributes well (e.g., the fence in Fig. 1), may lack the ability to capture the statistical attributes (e.g., the grass in Fig. 1). To alleviate this problem, we propose a novel ranking layer which can be embedded in the structure of classical CNN to form the Ranking-CNN. A Ranking-CNN can capture the structural and statistical attributes simultaneously. As a straightforward method, an end-to-end regression network can be established to estimate the transmission of each pixel from its surrounding local patch. However, since the regression target is only a real value between $(0, 1]$, when training the network using backward propagation algorithm, the gradient may be small and not robust. Therefore, it is difficult to effectively train the deep network. To this end, the regression problem is converted into a classification problem. Then the Ranking-CNN can be effectively trained on massive hazy image patches, and various types of haze-relevant features can be automatically learned. Based on these features, the random forest is further adopted to train a regression model so as to predict the transmission. Experimental results on plentiful synthetic and real-world images show that the proposed approach outperforms several previous outstanding approaches.

The main contributions of this paper include: First, we propose a novel ranking layer as well as its forward and backward computations, and theoretical analyses illuminate its excellent ability to capture statistical attributes. Second, by incorporating the ranking layer into the classical CNN, we construct a Ranking-CNN to learn effective haze-relevant features, which demonstrates impressive performance in image dehazing. Third, we benchmark the proposed dehazing approach and several state-of-the-art methods on extensive qualitative and quantitative experiments, in which the proposed approach achieves satisfactory performance.

The rest of this paper is organized as follows. Section II presents some related works. Section III formulates the problem and overviews our pipeline. Then each step is detailedly explained in Section IV. Finally we show the experimental results in Section V and conclude this paper in Section VI.

II. Related Work

In the past two centuries, the interaction phonomenon of light with the atmosphere has been widely studied [25], [27], [47], which is known as atmospheric optics. Based on the physical phonomenon, depending on whether using additional information, there are mainly two kinds of image dehazing methods. We then review the related works from this perspective. In addition, we also briefly introduce several representative works on deep neural network.

Image dehazing with additional information: Early methods usually use additional information to dehaze images. Nayar and Narasihan [29], [30] restore the scene structure from multiple images captured under different weather conditions, then the clear image can be recovered. Schechner *et al.* [38] observe that the scattered atmospheric light is usually partially polarized, then they take two or more images through a polarizer at different orientations for image dehazing. Shwartz *et al.* [41] automatically recover the parameters of the atmospheric light needed by polarizer based image dehazing methods. Kopf *et al.* [19] use the geometry of the scene to dehaze image via registering the hazy image into 3D scenes manually. However, as these additional information is usually difficult to obtain, these methods have many limitations.

Single image dehazing: As single image dehazing is an ill-posed problem, various priors and hypotheses have been proposed to tackle this problem. Oakley and Bu *et al.* [31] assume a constant air-light and estimate it via finding the minimum of a global cost function. Tan [44] removes the haze layer based on the observations that clear images have more contrast and the transmission tends to be smooth. Fattal [9] assumes that the shading and transmission functions are locally statistically uncorrelated. Tarel and Hautière [46] propose a fast algorithm whose complexity is a linear function of the image size. Kratz and Nishino [20] assume that the albedo and depth are statistically independent, then formulate a factorial Markov random field to estimate the transmission. He *et al.* [14] observe that the lowest value of each channel in a local image patch tends to be zero for clear images, which called dark channel prior. Wen *et al.* [51] further develop the underwater dark channel prior for image enhancement. Gibson *et al.* [13] investigate the dehazing effects on image and video coding. They further [12] use locally adaptive Wiener filter to refine the estimated density of haze. Yan *et al.* [55] reduce the amplified noise in the dehazed result image restored from dense haze. Fattal [10] utilizes the color-lines prior in local image patch. Sulami *et al.* [42] apply the color-lines prior to estimate an appropriate global constant atmospheric light vector. Wang and Fan [50] propose a multi-

scale depth fusion (MDF) method with local Markov regularization to blend multi-level details of chromaticity priors. Zhu *et al.* [57] propose a color attenuation prior and further apply a linear model for haze removal. Wang *et al.* [49] propose a fast method based on linear transformation. For each prior, it can be applied to a range of hazy images, however, there are often images which may not meet it. To this end, this paper aims at automatically learning information from massive data.

Recently, there are several learning-based image dehazing methods. Tang *et al.* [45] train a regression model to estimate the transmission via incorporating four types of haze-relevant features. Two recent works [4], [32] also adopt CNN to perform image dehazing. Ren *et al.* [32] directly estimate the whole transmission map from an input image via multi-scale CNN under the FCN framework [24]. Cai *et al.* [4] use a regression network to estimate the transmission of each pixel from its surrounding patch. However, these works mainly exploit existing hand-crafted features or classical CNNs. In contrast, we propose a novel Ranking-CNN to simultaneously capture statistical and structure attributes, which both are essential for single image dehazing.

Deep neural networks: Deep neural networks, also well known as deep learning or feature learning, are more powerful than shallow learning algorithms [17]. Many researchers use deep learning to perform high level computer vision tasks and significantly improve the performance, such as image classification [16], [21], object detection [7], [43], and semantic labelling [5], [8], [24]. Researches also have applied deep neural network to tackle low level problems and obtain promising results. Xie *et al.* [53] propose the Stacked Sparse Denoising Autoencoders (SSDA) to perform image denoising and inpainting. Agostinelli *et al.* [1] further propose adaptive multi-column stacked sparse denoising autoencoder (AMC-SSDA) to tackle multiple types of noise. Schuler *et al.* [39] train a multi-layer perceptron to perform image deconvolution task and obtain satisfactory results. Cho *et al.* [6] applies CNN on image matting. And Shen *et al.* [40] focus on portrait matting. These works demonstrate that the deep neural network can achieve satisfactory results not only on high-level problems but also low-level problems.

III. Overview

To dehaze an image, we first briefly formulate the formation process of a hazy image. Under the hazy or foggy weather, the scene radiance is scattered by the small particles suspending in the atmosphere. With increasing scene depth, the camera sensor captures less scene radiance but more atmosphere light. Thus, the formation of a hazy image can be described as a convex combination of the scene radiance $\mathbf{J}$ and the atmospheric light $\mathbf{A}$, which can be formulated as [30]

$$\mathbf{I}(x) = \mathbf{J}(x)\, t(x) + \mathbf{A}(x)\,(1 - t(x)), \tag{1}$$

where $\mathbf{I}(x)$ is a pixel from the hazy image $\mathbf{I}$ and $t(x)$ is its transmission. As a consequence, the problem of single image dehazing can be described as recovering the scene radiance $\mathbf{J}(x)$ from the hazy pixel $\mathbf{I}(x)$. From (1), we have

$$\mathbf{J}(x) = \frac{\mathbf{I}(x) - \mathbf{A}(x)\,(1 - t(x))}{t(x)}. \tag{2}$$

Note that the dehazing process in (2) is ideal and may require slight variations in building the computational model for image dehazing. From (2), we find that the dehazing problem can be decomposed to three subproblems, including:

1) Estimate the atmospheric light $\mathbf{A}(x)$,
2) Predict the transmission $t(x)$,
3) Recover scene radiance $\mathbf{J}(x)$.

To address these subproblems, the system framework of our approach is shown in Fig. 2. Specifically, since transmission prediction is often considered to be the key and most challenging subproblem in image dehazing [10], [14], [45], we propose the Ranking-CNN for this subproblem. Similar to the solutions in [14], [45], we also assume that the atmospheric light is constant for all image pixels. Then, we calculate the dark channel of the input hazy image using the approach in [14], and the atmospheric light $\mathbf{A}(x)$ at any pixel x is estimated by averaging the RGB color of the 0.1% pixels with the largest dark channel values.

Once the atmospheric light is estimated, we only have to focus on predicting the transmission $t(x)$ for every pixel according to its local features. To extract haze-relevant features, the proposed Ranking-CNN extends the structure of the classical CNN by adding a novel ranking layer so that the statistical and structural attributes of hazy image patches can be simultaneously captured. Based on the haze-relevant features, a transmission prediction model is then trained using the random forest regressor. The random forest regressor is adopted due to its several advantages, such as it can measure the importance of features and avoid seriously over-fitting. This regression model can be used to obtain the initial transmission for every pixel in the input image. To avoid edge artifacts, a guided filter is applied to refine the initial transmission, and the refined transmission is combined with the estimated global atmospheric light for image dehazing. As the Ranking-CNN model and the regression model are trained on massive amounts of data, they are effective for different input hazy images. Thus, we only need to train one unique Ranking-CNN model and one unique regression model, which are then used to dehaze any input hazy image.

IV. The Approach

In this section, we first introduce what the ranking layer is and how to add it to the structure of the classical CNN so as to construct the Ranking-CNN. After that, we describe the implementation details of the Ranking-CNN and show how to learn haze-relevant features. Finally, we demonstrate how to dehaze an input image with the features extracted by the Ranking-CNN.

A. Ranking Layer

By analysing the mechanism of existing image dehazing methods, we find that two types of attributes may influence the performance of transmission estimation, including statistical attributes (e.g., dark channel prior [14] and color-lines prior [10]) and structural attributes (e.g., boundaries [26]). Inspired

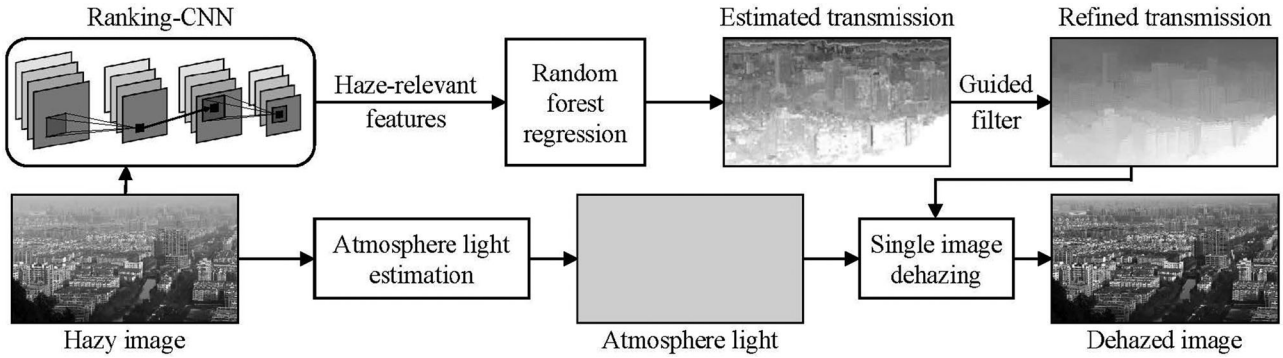


Fig. 2. System framework of our approach. Given a hazy image, we first estimate a global atmospheric light and use a pre-trained Ranking-CNN to extract haze-relevant features for each pixel from its surrounding patch. After that, the initial transmission is estimated via a random forest regression model, which is then refined through a guided filter. Finally, the clear image is recovered through single image dehazing.

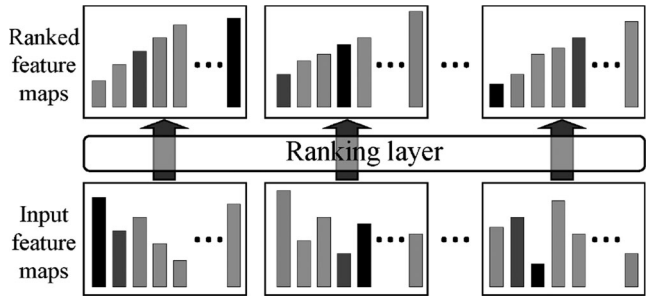


Fig. 3. A ranking layer operates separately on each input feature map and only changes the ordering of elements in each feature map other than modifying their values. Note that a feature map is actually a 2D matrix and here we turn them into a 1D vector by sampling elements column-wise so as to provide a better viewing experience.

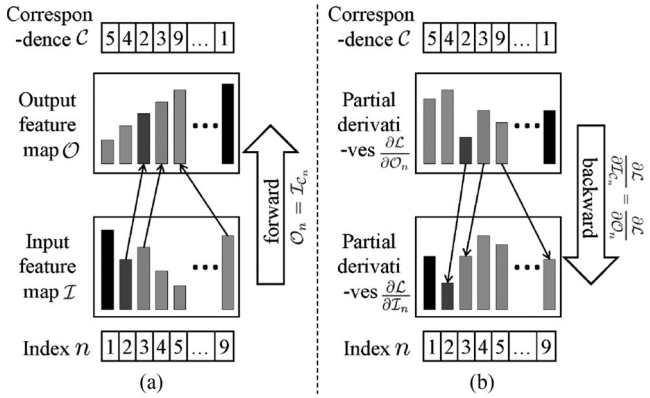


Fig. 4. The forward and backward propagation of the ranking layer on a specific feature map. In the forward propagation: the ranking layer sorts all the elements in a feature map and records the correspondence $\mathcal{C}$ between the input and output feature maps. In the backward propagation: the ranking layer propagates the partial derivatives from the output feature map to the input feature map according to the correspondence $\mathcal{C}$.

by this observation, we propose to automatically learn haze-relevant features through CNN so as to simultaneously capture these two types of attributes. However, CNN performs impressively on capturing the structural attributes due to the usage of convolutional layers, while it often lacks the ability to extract statistical attributes. Thus it is necessary to modify the structure of the classical CNN so as to enhance its ability in extracting haze-relevant features. Toward this end, we propose to add a ranking layer to the classical CNN so as to construct a novel Ranking-CNN.

For a ranking layer, its input consist of a number of feature maps, which is the same as a common layer of classical CNNs. The proposed ranking layer retains the values of all the elements in a feature map and only changes their ordering. The input of a ranking layer consists of a set of feature maps, and the ranking layer operates separately on each input feature map and output a ranked feature map with the same dimension (as shown in Fig. 3). Let $\mathcal{I}$ be an input feature map with N elements and $\mathcal{O}$ be its ranked version, we denote the nth element of $\mathcal{I}$ and $\mathcal{O}$ as $\mathcal{I}_n$ and $\mathcal{O}_n$, respectively. As shown in Fig. 4(a), in the forward propagation of a ranking layer, the element $\mathcal{O}_n$ corresponds to the nth smallest element in $\mathcal{I}$, whose index is denoted as $\mathcal{C}_n$, *i.e.*, $\mathcal{O}_n = \mathcal{I}_{\mathcal{C}_n}$. To facilitate the operations in the backward propagation, we record such pair-wise correspondences between the elements of input and output feature maps as $\{(\mathcal{C}_n, n)|1 \leq n \leq N\}$.

Based on the pair-wise correspondences $\{(\mathcal{C}_n, n)|1 \leq n \leq N\}$ between the elements of an input feature map and its output ranked version, the backward propagation at the ranking layer can be conducted. As the ranking layer only changes the ordering of elements in each feature map, the partial derivatives of the loss function $\mathcal{L}$ with respect to each output feature $\mathcal{O}_n$ can be directly passed to its corresponding input feature $\mathcal{I}_{C_n}$ as

$$\frac{\partial \mathcal{L}}{\partial \mathcal{I}_{\mathcal{C}_n}} = \frac{\partial \mathcal{L}}{\partial \mathcal{O}_n}. \tag{3}$$

In Fig. 4(b), we pick a specific feature map and visually explain the backward propagation of the ranking layer. Note that the ranking layer is parameter-free. No parameter needs to be learned in the backward propagation other than passing the derivatives.

The ranking layer operates separately on each feature map and sorts the elements in an input feature map in ascending order. Since the output feature map is ordered, extracting its statistical attributes, e.g., its contrast, becomes easier. As shown in Fig. 6(a), for various feature maps, classical CNN may need different convolutional filters (Fig. 6(b)) to compute the contrast.

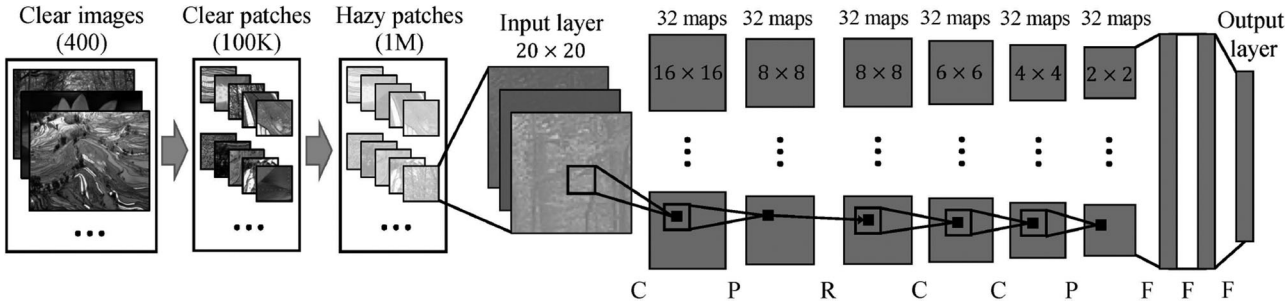


Fig. 5. The generation of training data and the structure of the Ranking-CNN. One million training patches are synthesized via adding random haze to 100 k clear image patches sampled from 400 clear images. The Ranking-CNN is constructed by adding a ranking layer to the structure of classical CNN (C: convolution; P: max pooling; R: ranking; F: fully-connected).

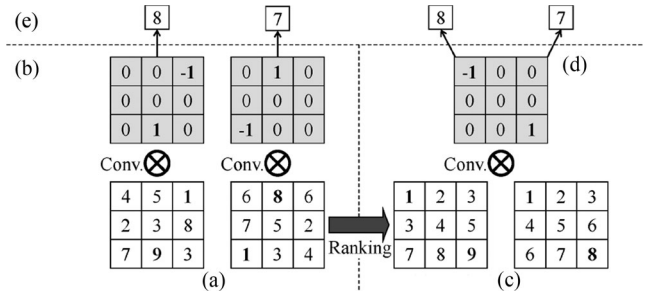


Fig. 6. An example to show how the ranking layer facilitates the statistical attributes extraction, e.g., the contrast. For (a) various feature maps, classical CNN may need (b) different convolutional filters to compute the contrast. However, if (c) the feature maps are ranked, only (d) one unique filter is needed.

However, if every feature map is ranked (Fig. 6(c)), only one unique convolutional filter (Fig. 6(d)) is needed to compute the contrast. As a whole, a ranked feature map facilitates its *statistical attributes* extraction. While the value of each feature, which is actually computed through classical convolutional or pooling operations, still reserves the *structural attributes*.

We finally analyse the computational complexity of the ranking layer. The computational complexity of the forward propagation is $O(n\lg(n))$ serially and $O(\lg(n))$ parallelly, since it acctualy performs a sort operation. The computational complexity of the backward propagation is $O(n)$ serially and $O(1)$ parallelly, since it directly propagates the derivatives according to the correspondence $\mathcal{C}$.

B. Learning Haze-Relevant Features

Given the ranking layer and the CNN, three issues still need to be addressed to learn haze-relevant features, including: 1) generating training data; 2) determining the structure of Ranking-CNN; 3) optimizing the parameters of Ranking-CNN.

Due to the lack of large-scale benchmarks, it is difficult to collect sufficient training data. Thus we address the first issue by generating massive synthesized hazy image patches for training the Ranking-CNN. As shown in Fig. 5, we first collect 400 clear images from the Internet, including various types of scenes, such as mountain, forest, grass, city, building, street scene, etc. From these images, we randomly select 100,000 clear image patches with the resolution 20×20. Based on these patches, we follow the formation process of a hazy image in (1) to generate massive hazy patches. Given a clear patch $\mathcal{B}$, we choose 10 random transmission $t_{\mathcal{B}}$ between $(0, 1]$ and assume that the transmission on each small image patch is constant. Thus the hazy patches can be synthesized via simulating the formation process of hazy images in (1). Since the main objective of Ranking-CNN is to learn haze-relevant features for transmission prediction, we use the same atmosphere light for all patches in the synthesization process (*i.e.*, $(1, 1, 1)^{\mathrm{T}}$). Finally, we have 1,000,000 synthesized hazy patches for learning haze-relevant features.

Before training the Ranking-CNN, we have to determine its structure. As shown in Fig. 5, our Ranking-CNN has ten layers. The first layer is the input layer, which includes the RGB channels of a color image patch with resolution 20×20. The second layer is a convolutional layer, where the R, G, B maps are convolved with 5×5 convolutional kernels to generate 32 feature maps with resolution 16×16. The third layer is a max pooling layer that sub-samples the input feature map over each 2×2 non-overlapping window. The fourth layer is the ranking layer, which operates separately on each input feature map. It sorts all the elements in an input feature map and outputs a ranked feature map with the same dimension. Note that the elements in the ranked feature map are in ascending order from left-top to right-bottom. The fifth layer is a convolutional layer, which includes 32 feature maps and the convolutional kernel size is 3×3. The sixth layer is also a convolutional layer same with the fifth layer. The seventh layer is another max pooling layer which is the same as the third layer. After this layer, we finally obtain 32 feature maps of size 2×2. The eighth layer, ninth layer (each with 64 features) and the output layer (with 10 output values) are all fully-connected layers. In our Ranking-CNN, we use rectified linear unit (ReLU) activation function [28] for all convolutional layers and the first fully-connected layer. For each hazy patch $\mathcal{B}$, the 10D output vector (denoted as $\mathbf{Y}_{\mathcal{B}}$) are expected to approximate the label vector $\mathbf{N}_{\mathcal{B}} = \left(n_{\mathcal{B}}^1, n_{\mathcal{B}}^2, \ldots, n_{\mathcal{B}}^{10}\right)^{\mathrm{T}}$, where $n_{\mathcal{B}}^i \in \{0, 1\}$ is a binary variable that can be calculated as

$$n_{\mathcal{B}}^i = \begin{cases} 1, & \text{if } t_{\mathcal{B}} \in (i/10 - 0.1, i/10] \\ 0, & \text{otherwise} \end{cases}. \tag{4}$$

In other words, we treat the Ranking-CNN as a multi-class classifier and try to optimize its parameters via maximizing the classification accuracy.

Intuitively, we can train an end-to-end network that predicts the transmission by replacing the output layer with a linear regression layer. However, since the output of the linear regression layer is only a variable varying between $(0, 1]$, it is difficult to effectively train the deep network. To facilitate the training process, we adopt a two-stage training scheme. That is, we first convert the problem to a 10-category classification problem and train a Ranking-CNN model for classification. After that, the output layer is discarded and the output of the second fully-connected layer is used as features for training a random forest regressor to predict the transmission. In this manner, the training process of the Ranking-CNN is easier and the learned features, when they are combined with the random forest regressor, still have impressive performance in image dehazing.

To train the Ranking-CNN model, we minimize a soft-max loss function to optimize the parameters in the network. The loss function is defined as

$$\mathcal{L}\left(\mathbf{N}_{\mathcal{B}}, \mathbf{Y}_{\mathcal{B}}\right) = -\log\left(\frac{e^{y_{\mathcal{B}}^{j}}}{\sum_{i=1}^{10} e^{y_{\mathcal{B}}^{i}}}\right), \tag{5}$$

where $y_{\mathcal{B}}^{i}$ is the ith element of $\mathbf{Y}_{\mathcal{B}}$, and j is the index that $n_{\mathcal{B}}^{j} = 1$. To optimize the parameters in Ranking-CNN, we use the back-propagation algorithm with stochastic gradient descent solver [22]. We set the initial learning rate r_{l_0} as 0.01, the momentum as 0.9, the mini-batch size as 64. As shown in previous literatures [4], [32], it is helpful to decrease the learning rate along with the training process. Therefore, we update the learning rate as

$$r_l = r_{l_0} \times (1 + 0.0001 \times iter)^{-0.75}, \tag{6}$$

where $iter$ is the index of training iteration on each mini-batch. In the experiments, we perform 100 epoches on the whole training data, and the 64D output of the second fully-connected layer are used for transmission prediction.

C. Image Dehazing

Based on the learned features, we further use the Random Forest [3] to learn a regression model between the transmission t and the haze-relevant features. Our random forest model has 200 trees and each tree random selects $1/3$ feature dimensions. For efficiency, we random select $1/100$ hazy image patches (*i.e.*, 10,000) to train the regression model. Note that we set the atmospheric light as a constant vector (*i.e.*, $(1, 1, 1)^{\mathrm{T}}$) during training process. To relax this condition, we first apply white balance on the input image using our estimated atmospheric light $\mathbf{A}$. In our approach, white balance is applied by dividing each channel c of the input image by the corresponding channel of the estimated $\mathbf{A}$ as

$$\mathbf{I}'^{c}(x) = \frac{\mathbf{I}^{c}(x)}{\mathbf{A}^{c}(x)} = \frac{\mathbf{J}^{c}(x)}{\mathbf{A}^{c}(x)} t(x) + (1 - t(x)). \tag{7}$$

Thus $\mathbf{I}'$ can be regarded to have atmospheric light of $(1, 1, 1)^{\mathrm{T}}$ and the transmissions of $\mathbf{I}'$ and $\mathbf{I}$ are the same.

In the training data synthesizing process, we also assume that the transmission coefficients are locally consistent. However, we do not hold this assumption in the dehazing process. Therefore, we extract the haze-relevant features for every pixel in input image via selecting a 20×20 patch centred at the pixel using the Ranking-CNN. With these features, the regression model is applied to estimate the transmission $t(x)$ for each pixel x. To avoid the artifacts near object edges, we further use guided filter [15] to smooth the initial estimated transmission for efficiency. Laplacian matting [23] also can be used instead to get more satisfactory results around edges. After obtaining the transmission $t(x)$ and atmosphere light $\mathbf{A}(x)$ for each pixel x, we can dehaze the input image by applying the ideal dehazing process in (2). Moreover, to avoid the strong fluctuation of recovered pixel when the transmission is very small, we set $t(x) = 0.05$ if $t(x) < 0.05$. Thus we can get the clear image as

$$\mathbf{J}(x) = \frac{\mathbf{I}(x) - \mathbf{A}(x)}{\mathbf{max}(t(x), 0.05)} + \mathbf{A}(x). \tag{8}$$

As the exposure is determined according to the hazy scene, the dehazed image usually tends to be underexposure, *i.e.*, the luminance $\mathbf{J}^{l}(x)$ of $\mathbf{J}$(x) is usually much less than the luminance $\mathbf{I}^{l}(x)$ of $\mathbf{I}(x)$. Therefore, we adaptively increase the exposure as $\mathbf{J}^{*}(x) = \lambda \mathbf{J}(x)$, where $1 \leq \lambda \leq \frac{\sum_x \mathbf{I}^{l}(x)}{\sum_x \mathbf{J}^{l}(x)}$ is the exposure factor. As there are many regions which tend to be gray in the input hazy image, the dehazed image will be overexposure if $\lambda = \frac{\sum_x \mathbf{I}^{l}(x)}{\sum_x \mathbf{J}^{l}(x)}$. As a compromise, the log function is used in our method and

$$\lambda = \log\left(\frac{\sum_x \mathbf{I}^{l}(x)}{\sum_x \mathbf{J}^{l}(x)}\right) + 1. \tag{9}$$

Then, the exposure can be increased and overexposure also can be avoided at the same time.

V. Experiments

We first compare the dehazed results of our method and several previous methods on both synthetic and real benchmark images. Then we exploit the influence of the ranking layer and compare the features learned by our Ranking-CNN with previous haze-relevant features quantitatively.

A. Comparisons With Previous Approaches

The dehazed results and comparisons can be found in Figs. 7–9, which are achieved respectively on synthetic hazy images with ground-truth clear images and transmissions, captured hazy images with known clear images, and real benchmark hazy images without ground-truth. The experimental results show that, our method can achieve better results compared with several previous methods both quantitatively and qualitatively. In our experiments, we implement our Ranking-CNN to learn haze-relevant features based on the open source deep learning framework Caffe [18]. We reimplement the methods of [14] and [45], and directly use the published results or codes of other referenced methods, such as [2], [4], [10], [32], [57].

In order to perform quantitative comparison, like some previous methods [4], [32], [45], we synthesize ten hazy images based on stereo benchmark images published in [35]–[37], which is denoted as Dataset-Syn. Each image from this dataset has two types of ground-truth, including a haze-free image and a ground-truth transmission map. To be fair, we follow the experiments

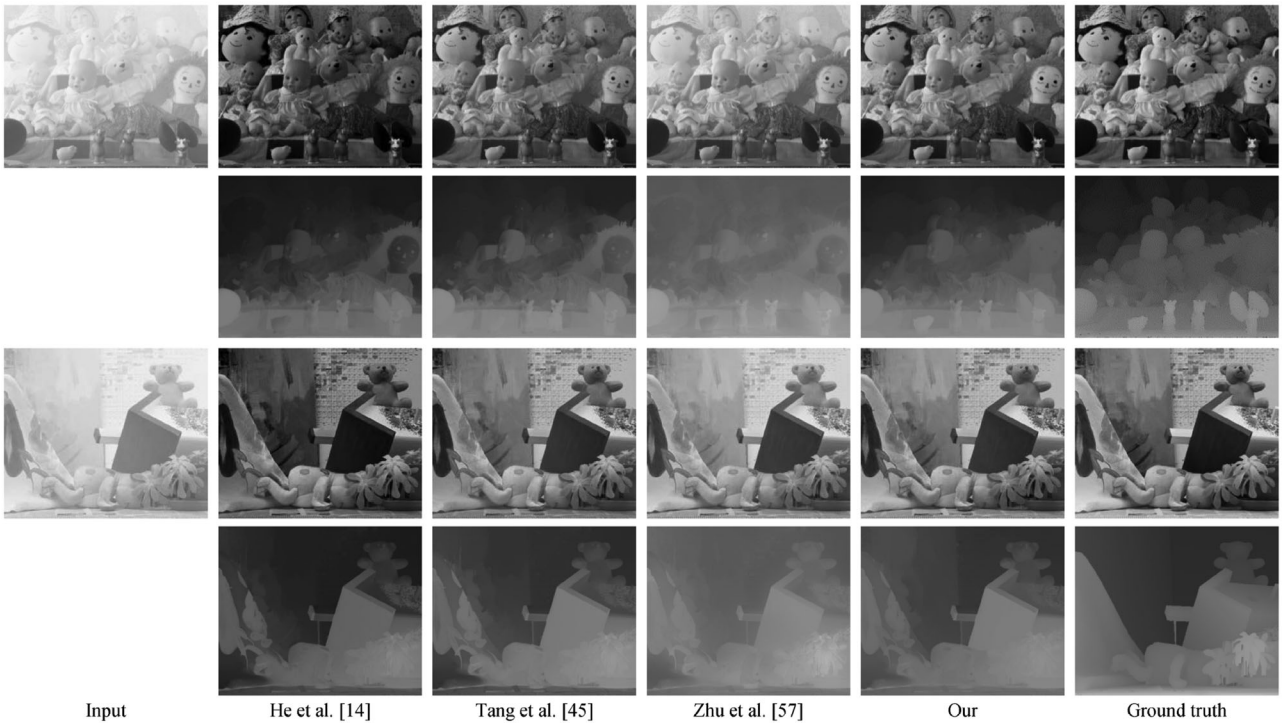


Fig. 7. Representative dehazed results on Dataset-Syn. We can see that, Zhu *et al.* [57] usually under estimate the transmission, while [14] and [45] usually over estimate the haze, such as the light pink pig, the light brown heads, the red teddy and the gray areas.

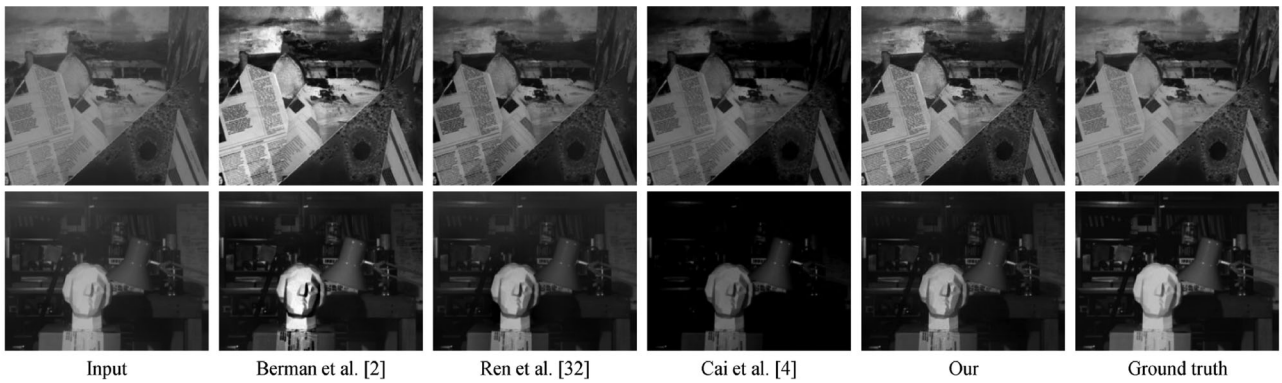


Fig. 8. Representative dehazed results on Dataset-Cap. We can see that our method can achieve satisfactory results on images with light haze. These results illustrate the robustness of our method.

set-up in [45] and set the transmission $t(x) = 0.8 \times d(x)$ for each pixel x, where $d(x)$ is the disparity. Table I shows the L_1 error comparisons in transmission and image of our method and [2], [4], [14], [32], [45], [57]. The L_1 error in transmission is calculated between the estimated and ground-truth transmission maps, and the L_1 error in image is calculated between the dehazed and haze-free images. Overall, as can be seen, our method achieves the best results and has over 10% lower average L_1 error in estimated transmission and dehazed image compared with these methods. There are two dehazed results illustrated in Fig. 7, we can see that [14] usually over estimate the haze, such as the light pink pig, the light brown heads, the red teddy and the gray areas. [45] also suffers this problem as the multi-scale dark channel features are the most important features in their method. On the contrary, our method suffers less over estimated problems.

Though the Dataset-Syn are synthesized following the abstractly formulation of hazy images (1), however, the physical process may not follow it precisely. To this end, inspired by the construction of image matting benchmark [33], [34], we design a process to directly capture a hazy image as well as its corresponding clear version. We first use a Lenovo 22" monitor to display each clear image in Dataset-Syn, and capture it by a Cannon 650D DSLR camera. After that, an ultrasonic humidifier is used to fill vapour between the monitor and the camera. Then the camera captures the hazy image with all the other settings and parameters unchanged. Except the vapour, the environment settings and camera parameters are unchanged, therefore the

Fig. 9. Representative results obtained by our approach and previous methods. The results show that, our method can achieve visual better results on a lot of real benchmark images. Specially, our method suffers less over estimating problems or color shifts, such as the faces of the two actresses, and the green trees.

captured clear image can be regarded as the ground-truth of its corresponding captured hazy image. This captured dataset is denoted as Dataset-Cap. Table II shows the L_1 error comparisons in image of our method and [2], [4], [14], [32], [45], [57]. Our method still achieves the best results. There are two dehazed results illustrated in Fig. 8. As Figs. 7 and 8 show, the hazy of Dataset-Syn is dense and it of Dataset-Cap is light. Our method can achieve the best performance on both these two datasets, which means that our method performs robust under different hazy density than other methods.

Finally, we conduct a subjective test to visually compare the results of our method, [14], [45], [10], [2], [4] and [32] on 69 benchmark images. For a fair comparison, we use the results which are published by [14], [45], [10], and generate the results using the codes which are published by [2], [4], [32]. Since each paper only publishes results on a portion of the 69 images, we

TABLE I
THE L_1 ERRORS ON STEREO DATASET-SYN

	He *et al.* [14]	Tang *et al.* [45]	Zhu *et al.* [57]	Berman *et al.* [2]	Ren *et al.* [32]	Cai *et al.* [4]	Ours
Aloe	0.100 / 0.191	0.060 / 0.087	0.175 / 0.141	0.060 / 0.086	–/ 0.195	0.089 / 0.096	**0.051** / **0.070**
Art	0.116 / 0.176	0.077 / 0.098	0.114 / 0.145	0.099 / 0.123	–/ 0.210	0.094 / 0.122	**0.061** / **0.076**
Barn	0.079 / 0.089	0.061 / 0.063	0.075 / 0.079	0.128 / **0.049**	–/ 0.174	0.075 / 0.079	**0.051** / 0.055
Bull	0.050 / 0.122	0.035 / 0.091	0.184 / 0.265	0.049 / 0.102	–/ 0.337	0.110 / 0.202	**0.023** / **0.061**
Cones	0.084 / 0.102	0.043 / 0.044	0.106 / 0.110	0.055 / 0.071	–/ 0.178	0.081 / 0.087	**0.034** / **0.036**
Dolls	0.061 / 0.110	0.038 / 0.069	0.152 / 0.201	0.067 / 0.095	–/ 0.272	0.076 / 0.132	**0.032** / **0.060**
Flower	0.059 / 0.105	0.046 / **0.066**	0.146 / 0.172	0.066 / 0.145	–/ 0.239	0.098 / 0.135	**0.045** / 0.068
Teddy	0.092 / 0.135	0.055 / **0.060**	0.124 / 0.126	0.092 / 0.125	–/ 0.167	0.082 / 0.089	**0.054** / 0.061
Tsukuba	0.068 / 0.093	0.077 / **0.123**	0.173 / 0.253	**0.060** / 0.113	–/ 0.329	0.117 / 0.182	0.077 / 0.125
Venus	0.042 / 0.074	0.046 / 0.103	0.159 / 0.239	0.051 / 0.163	–/ 0.310	0.114 / 0.196	**0.035** / **0.079**
Average	0.075 / 0.120	0.053 / 0.080	0.141 / 0.173	0.073 / 0.107	–/ 0.241	0.094 / 0.132	**0.046** / **0.069**

Left values indicate L_1 error in transmission. Right values indicate L_1 error in image.

TABLE II
THE L_1 ERRORS ON DATASET-CAP

	He *et al.* [14]	Tang *et al.* [45]	Zhu *et al.* [57]	Berman *et al.* [2]	Ren *et al.* [32]	Cai *et al.* [4]	Ours
Aloe	0.169	0.175	**0.091**	0.130	0.169	0.313	0.134
Art	0.136	0.087	0.073	0.090	0.079	0.231	**0.064**
Barn	0.054	0.046	0.070	0.087	0.061	0.117	**0.041**
Bull	0.064	0.075	**0.046**	0.065	0.087	0.206	0.053
Cones	0.093	0.064	**0.053**	0.057	0.104	0.213	0.057
Dolls	0.103	0.074	0.066	0.083	0.089	0.201	**0.057**
Flower	0.070	0.049	0.052	0.080	0.068	0.212	**0.037**
Teddy	0.126	0.108	**0.067**	0.141	0.129	0.197	0.089
Tsukuba	0.065	0.060	0.072	0.057	0.655	0.259	**0.048**
Venus	**0.048**	0.059	0.053	0.105	0.071	0.190	0.050
Average	0.093	0.080	0.064	0.089	0.092	0.214	**0.063**

Values indicate L_1 error in image.

obtain 1012 pairs of dehazed results in total. Fifteen subjects are invited to perform this experiment. All these subjects have normal or corrected normal visual acuity and normal color vision. The results are shown on a normal $22''$ display with 1680×1050 resolution. The display is placed in a room with fluorescent lamps. On each of such pair-wise comparisons, four images are shown in a 2×2 grid. The top-left is the input hazy image. The top-right is the ground truth hazy free image (if it exists, otherwise the input hazy image). The bottom-left and bottom-right are the dehazed results from two methods, each result is random shown in left or right. Each image is shown with no more than 640×480 resolution, which also can be shown with its original resolution in a new window by click. Each subject is requested to observe each comparison and determine which dehazed result is better. Averagely, each subject takes about 75 minutes to perform the test. Note that the methods that are being compared are blind to the subjects. Among all these $1012 \times 16 = 16192$ pair-wise comparisons, our method achieves the first place and outperforms the other methods for 3221 times, while [14] takes the second place (2649 times). These results, together with the objective performance, indicate that our method performs the best in both objective and subjective experiments compared with the several referenced methods.

We also show some dehazed results on real world images in Fig. 9. It shows that, our method can achieve visual better results on a lot of real benchmark images. Specially, our method suffers less over estimating problems and color shifts, such as the faces of the actresses and the green trees.

B. Performance Analysis

Beyond the performance comparisons, in this section we conduct a number of small experiments to validate the performance of our approach from multiple perspectives. For quantitatively evaluation, we further generate 400,000 hazy patches with synthetic transmission as validation set.

Features comparison: In the first experiment, we compare the performance of various types of features, including the 64D features learned by the Ranking-CNN (denoted as $\mathcal{F}_R$), the 64D features learned by the classical CNN (only remove the ranking layer in the Ranking-CNN and keep the other experimental setups unchanged, denoted as $\mathcal{F}_C$), the 325D features designed by [45] (denoted as $\mathcal{F}_T$) and the combination of $\mathcal{F}_T$ and $\mathcal{F}_R$ (denoted as $\mathcal{F}_{T+R}$). Moreover, the Ranking-CNN model and classical CNN model are respectively obtained after 100 epoches on the identical training set. The parameters of each layer are initialized via xavier method. The 325D features of $\mathcal{F}_T$ consist of multi-scale dark channel priors and local max contrasts, hue disparity and multi-scale local max saturation. For efficiency issue, we random select $1/100$ training patches (*i.e.*, 10,000) from that are used by the Ranking-CNN to train the random forest regression model. Fig. 10 shows the L_1 error in transmission on validation set using different combinations of the features, *i.e.*

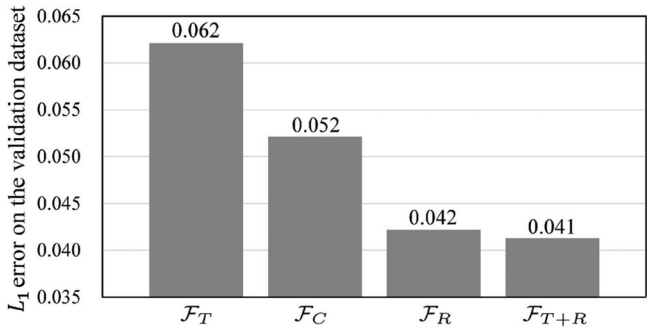


Fig. 10. The L_1 errors on validation set when different features are used for dehazing. $\mathcal{F}_T$: features used in [45]; $\mathcal{F}_C$: features learned by the classical CNN; $\mathcal{F}_R$: features learned by the Ranking-CNN; $\mathcal{F}_{T+R}$: the combination of $\mathcal{F}_T$ and $\mathcal{F}_R$.

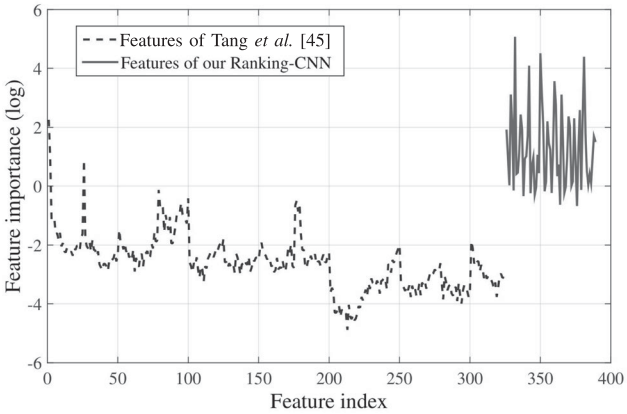


Fig. 11. The importance of features learned from Ranking-CNN and features heuristically designed in [45].

$\mathcal{F}_T$, $\mathcal{F}_C$, $\mathcal{F}_R$ and $\mathcal{F}_{T+R}$. We can see that the features from the Ranking-CNN outperform those from [45] by 32% in terms of L_1 error. Moreover, our Ranking-CNN features achieve about 20% better compared with the classical CNN features. If we combine our Ranking-CNN features with the features used by [45], the L_1 error is only decreased slightly (0.001), which means that our Ranking-CNN features not only capture most information in the previous hand-crafted features, but also learn more information from the massive data automatically. This experiment also shows that both structural features (e.g., CNN features) and statistical features (e.g., features used in [45]) are useful for transmission estimation.

We further explore the importance of each dimension in features $\mathcal{F}_{T+R}$, which consists of the 64D features from the Ranking-CNN and 325D features used in [45]. All these features are incorporated to train a random forest regressor, and the importance of each feature dimension can be obtained. As illustrated in Fig. 11, we plot the importance of each feature dimension which can be obtained from the trained random forest regressor. It is obviously that our Ranking-CNN features are more important than the previous features used in [45]. Moreover, the sum importance of the Ranking-CNN features is 708.48, while the sum importance of the previous features $\mathcal{F}_T$ is only 47.41, which shows that the Ranking-CNN features are powerful and remarkably outperform the previous heuristic designed features used in [45].

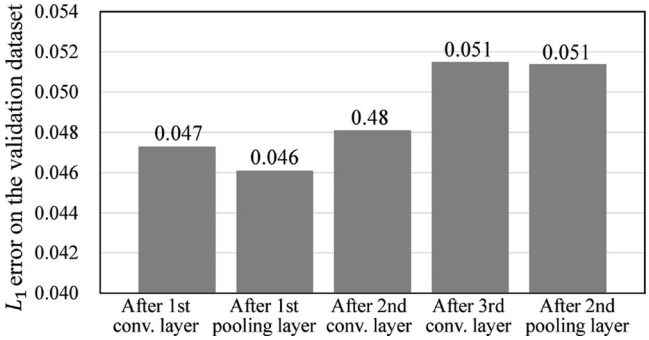


Fig. 12. The L_1 error in transmission on the validation set using our Ranking-CNN when the ranking layer is placed at different locations.

We also compare the features generated from different layers by training the regression model. The L_1 errors on the same validation dataset are 0.050, 0.043 and 0.042 by using the 128D features generated from the second pooling layer, the 64D features generated from the first fully-connected layer and the 64D features generated from the second fully-connected layer, respectively. This may imply that the features from deeper layers are more powerful. Thus we adopt the 64D features generated from the second fully-connected layer for transmission estimation.

The location of the ranking layer: In the third experiment, we show the performance of the Ranking-CNN when the ranking layer is placed at different locations. In the experiment, the ranking layer is placed after the first convolutional layer, the first pooling layer (as in Fig. 5), the second and third convolutional layer, and the second pooling layer respectively. Fig. 12 shows the L_1 error in transmission on the validation set respectively after 20 training epoches. We can see that when the ranking layer is placed at the fourth layer (after 1st pooling layer), our dehazing model achieves the minimal L_1 error. To explain this phenomena, we rethink the problem from another perspective: what features will be extracted without the ranking layer? As state in [56], the shallow layers of the classical CNN extract low-level features like boundaries and contrasts, while the deep layers extract high-level features like patterns and objects. In existing studies haze has been proved to be tightly correlated with low-level structural features like boundaries [26] as well as the statistical information like dark channel prior [14] and color-lines prior [10]. As the ranking layer only re-ranks the order of the features and keeps the values unchanged, the low-level structural informations of the input image still can be maintained in the Ranking-CNN. By inserting the ranking layer after the first pooling layer, the shallow layers before the ranking layer can make full use of the low-level structural features like boundaries. While additional statistical information also can be incorporated more easily after the ranking operations. By fusing both the low-level structural information and statistical information, the proposed method can achieve the best performance by placing the ranking layer at the fourth layer.

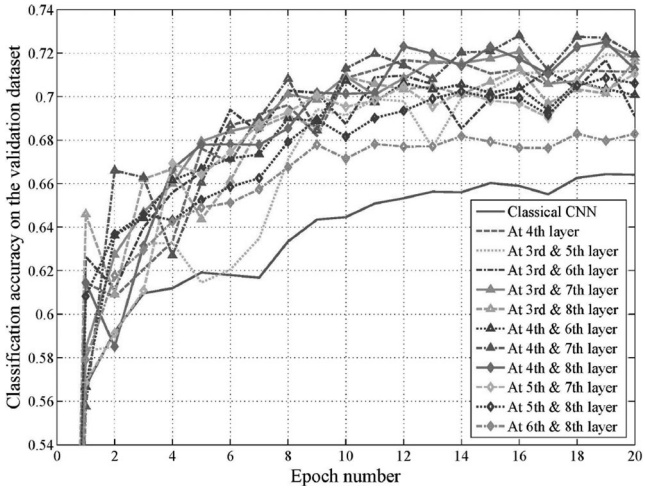


Fig. 13. The classification accuracy on the validation dataset when two ranking layers are placed at all possible locations in CNN.

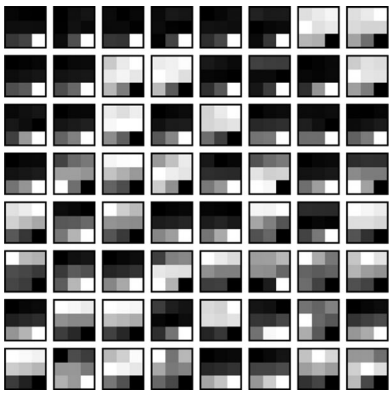


Fig. 14. The visualization of 64 filters randomly sampled from the second convolutional layer of the Ranking-CNN.

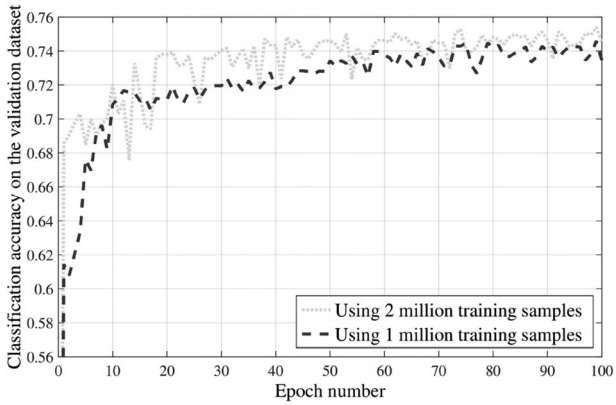


Fig. 15. The classification accuracy of two Ranking-CNN models on the same validation set, which are trained on one million or two million synthetic training samples, respectively. We can see that more training data generally bring better performance.

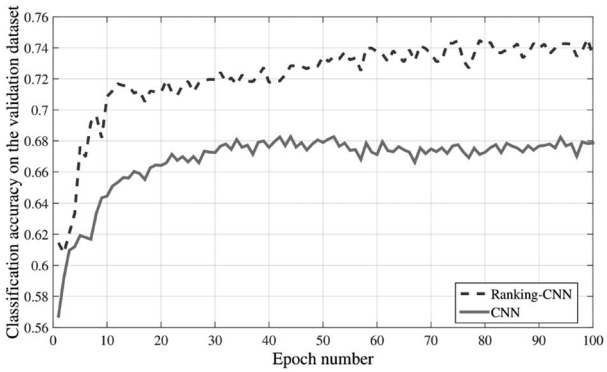


Fig. 16. The classification accuracy of our Ranking-CNN when different number of epoches are performed in the training process. Our Ranking-CNN can converge as quickly as classical CNN.

In the fourth experiment, we exploit the performance of the Ranking-CNN that uses two ranking layers. The two ranking layers are placed at all possible locations in the CNN. As shown in Fig. 13, the best performance is achieved by placing one ranking layer at the fourth layer (*i.e.*, after the first pooling layer) and the other one at the seventh layer (*i.e.*, after the second convolutional layer). However, the performance improvement, compared with the Ranking-CNN with only one ranking layer, is marginal (about 0.8% on the validation dataset, after 20 epoches). Considering the additional computational cost in the ranking layer, we still adopt one ranking layer for image dehazing.

Visualize the network: In the fifth experiment, we try to explain why the features learned by the Ranking-CNN are useful for image dehazing. We randomly select and visualize 64 filters from the second convolutional layer in Fig. 14. We can see that these filters actually provide cues on which elements in a local patch of a feature map should be referred to in extracting haze-relevant features. For instance, the filter at the left-top corner may imply that the largest value in a local patch should be considered for extracting haze-relevant features. It is somehow similar to the mechanism of dark channel prior, while the main difference is that various types of haze-relevant features are extracted by referring to different combinations of elements in a local patch. In this manner, the Ranking-CNN extracts an over-complete set of haze-relevant features, which are then weighted and selected in the random forest regressor. In this manner, Ranking-CNN demonstrates impressive performance in dehazing images.

The size of training data: In the sixth experiment, we explore the influence when different numbers of synthetic training data are used in training the Ranking-CNN model. As shown in Fig. 15, the accuracy of the Ranking-CNN on the same validation set increases about 1% after 100 epoches when 2 million training samples are used, while the training time is doubled as well. This result implies that our proposed method still has potential to be further improved by simply generating more synthetic training data. Considering the efficiency in the training stage, we use 1 million training samples in all the other experiments.

The convergence speed: In the seventh experiment, we compare the convergence speeds between the Ranking-CNN and the classical CNN. As shown in Fig. 16, the convergence speed of the Ranking-CNN is comparable to the classical CNN. This may

be caused by the fact that in the ranking layer the partial derivatives of the loss function with respect to each output feature can be directly passed to its corresponding input feature. Since the ranking layer is parameter-free, adding a ranking layer will not dramatically increase the difficulties in training the network.

Different regressors: In the second experiment, we test the performance of different types of regressors. Besides the random forest, we select three other regressors, including linear regressor, logistic regressor and SVM regressor(with radial basis function kernel). The L_1 errors of these regressors on the same validation dataset are 0.042 (random forest), 0.057 (linear), 0.054 (logistic) and 0.060 (SVM). As random forest regressor achieves the best performance, we employ it in transmission estimation.

The end-to-end method: To explore the performance of the end-to-end method, we replace the output layer of our Ranking-CNN by a linear regression layer. Then the modified Ranking-CNN can directly predict the transmission. However, though it is actually more efficient, the performance is unsatisfactory. In fact, its mean L_1 error on test data is 0.073, while our method achieves 0.042. The reason may be that, when we take the network as a regressor and train it to predict the transmission, the L_2 loss function is a common choice, which is also used in our experiment. When we train a classification network, we use the soft-max loss function. We can see that, the soft-max loss function is steeper than the L_2 loss function, which means that the classification network can be updated more effectively. Moreover, since the classification network outputs a number of probabilities about each label other than a real value, the learned features tend to be more various.

Running time: The last experiment is about the Running time. Our experiments are performed on a 3.1 GHz PC with a NVIDIA Geforce GTX980 GPU. Our feature learning and extracting algorithm is implemented based on Caffe. It takes about 400 seconds to perform one training epoch on all 1 million training samples using GPU. In feature extracting process, it takes about 283 seconds to extract features for 1 million patches, while classical CNN takes about 247 seconds. We use a C implementation of random forest and it takes about two minutes to train the regression model on our 10,000 training samples using CPU, and takes about 0.25 seconds to predict initial transmission of 10,000 patches. The other parts of our method are implemented using matlab, which takes several seconds for a typically 640×480 image. Our method achieve satisfactory performance quantitatively and qualitatively, the weakness is its efficiency. Compared with several previous methods, our method takes more time. The main reason is that we extract features and estimate the transmission for every pixel according its local patch. However, as the transmissions are correlated in a local patch, we can simultaneously estimate the transmissions of more pixels in the future work. Then, the running time can be decreased more than one order of magnitude.

VI. Conclusion

This paper presents a method to dehaze an image based on the features which are automatically learned from massive hazy images. To this end, a novel ranking layer is proposed to form the Ranking-CNN, that can learn haze-relevant features more effectively compared with the classical CNN. Equipped with the novel ranking layer, our Ranking-CNN can capture the structural and statistical features simultaneously. Based on the learned features, a regression model is further trained to predict haze density for effective haze removal. Experimental results show that our Ranking-CNN features are effective. The proposed image dehazing method, which is based on the features, also achieves satisfactory results on synthetic and real world data. At the same time, as we extract features for every pixel, the weakness of our method is its efficiency, which should be further improved in the future work, *i.e.*, via adopting FCN framework [24] to reduce redundant computations.

References

[1] F. Agostinelli, M. R. Anderson, and H. Lee, "Adaptive multi-column deep neural networks with application to robust image denoising," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 1493–1501.

[2] D. Berman, T. Treibitz, and S. Avidan, "Non-local image dehazing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1674–1682.

[3] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[4] B. Cai, X. Xu, K. Jia, C. Qing, 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.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," in *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017, to be published.

[6] D. Cho, Y.-W. Tai, and I. Kweon, "Natural image matting using deep convolutional neural networks," in *European Conference on Computer Vision* (Lecture Notes in Computer Science), vol. 9906. Berlin, Germany: Springer, 2016, pp. 626–643.

[7] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, Jun. 2014, pp. 2155–2162.

[8] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[9] R. Fattal, "Single image dehazing," *ACM Trans. Graph.*, vol. 27, no. 3, Aug. 2008, Art. no. 72.

[10] R. Fattal, "Dehazing using color-lines," *ACM Trans. Graph.*, vol. 34, no. 1, Dec. 2014, Art. no. 13.

[11] 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, Aug. 2016.

[12] K. B. Gibson and T. Q. Nguyen, "Fast single image fog removal using the adaptive wiener filter," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 714–718.

[13] K. B. Gibson, D. T. Vo, and T. Q. Nguyen, "An investigation of dehazing effects on image and video coding," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 662–673, Feb. 2012.

[14] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, Jun. 2009, pp. 1956–1963.

[15] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.

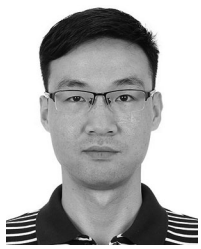
[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, Jun. 2016, pp. 770–778.

[17] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[18] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," *Proc. 22nd ACM Int. Conf. Multimedia*, Orlando, FL, USA, pp. 675–678.

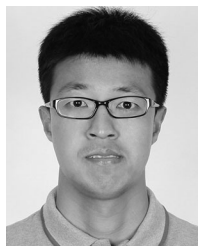
[19] J. Kopf *et al.*, "Deep photo: Model-based photograph enhancement and viewing," *ACM Trans. Graph.*, vol. 27, no. 5, Dec. 2008, Art. no. 116.

[20] L. Kratz and K. Nishino, "Factorizing scene albedo and depth from a single foggy image," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep. 2009, pp. 1701–1708.
[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[23] A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 228–242, Feb. 2008.
[24] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, Jun. 2015, pp. 3431–3440.
[25] E. J. McCartney, *Optics of the Atmosphere: Scattering by Molecules and Particles*. New York, NY, USA: Wiley, 1976.
[26] G. Meng, Y. Wang, J. Duan, S. Xiang, and C. Pan, "Efficient image dehazing with boundary constraint and contextual regularization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 617–624.
[27] W. E. K. Middleton, *Vision Through the Atmosphere*. Berlin, Germany: Springer, 1957.
[28] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.
[29] S. G. Narasimhan and S. K. Nayar, "Chromatic framework for vision in bad weather," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2000, vol. 1, pp. 598–605.
[30] S. K. Nayar and S. G. Narasimhan, "Vision in bad weather," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1999, vol. 2, pp. 820–827.
[31] J. P. Oakley and H. Bu, "Correction of simple contrast loss in color images," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 511–522, Feb. 2007.
[32] W. Ren *et al.*, "Single image dehazing via multi-scale convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis., Part II*, 2016, pp. 154–169.
[33] C. Rhemann, C. Rother, A. Rav-Acha, and T. Sharp, "High resolution matting via interactive trimap segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, Jun. 2008, pp. 1–8.
[34] C. Rhemann *et al.*, "A perceptually motivated online benchmark for image matting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, Jun. 2009, pp. 1826–1833.
[35] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, Jun. 2007, pp. 1–8.
[36] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, no. 1, pp. 7–42, 2002.
[37] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.,* Jun. 2003, vol. 1, pp. I-195–I-202.
[38] Y. Y. Schechner, S. G. Narasimhan, and S. K. Nayar, "Instant dehazing of images using polarization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2001, vol. 1, pp. I-325–I-332.
[39] C. J. Schuler, H. C. Burger, S. Harmeling, and B. Scholkopf, "A machine learning approach for non-blind image deconvolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, Jun. 2013, pp. 1067–1074.
[40] X. Shen, X. Tao, H. Gao, C. Zhou, and J. Jia, "Deep automatic portrait matting," in *European Conference on Computer Vision (Lecture Notes in Computer Science)*, vol. 9905. Berlin, Germany: Springer, 2016, pp. 92–107.
[41] S. Shwartz, E. Namer, and Y. Y. Schechner, "Blind haze separation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2006, vol. 2, pp. 1984–1991.
[42] M. Sulami, I. Geltzer, R. Fattal, and M. Werman, "Automatic recovery of the atmospheric light in hazy images," in *Proc. IEEE Int. Conf. Comput. Photography*, 2014, pp. 1–11.
[43] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, Jun. 2015, pp. 1–9.
[44] R. T. Tan, "Visibility in bad weather from a single image," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, Jun. 2008, pp. 1–8.
[45] K. Tang, J. Yang, and J. Wang, "Investigating haze-relevant features in a learning framework for image dehazing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, Jun. 2014, pp. 2995–3002.
[46] J.-P. Tarel and N. Hautière, "Fast visibility restoration from a single color or gray level image," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep. 2009, pp. 2201–2208.
[47] Y. M. Timofeev and A. V. Vasilév, *Theoretical Fundamentals of Atmospheric Optics*. Cambridge, U.K.: Cambridge Int. Sci. Publ., 2008.
[48] S. Wang *et al.*, "Guided image contrast enhancement based on retrieved images in cloud," *IEEE Trans. Multimedia*, vol. 18, no. 2, pp. 219–232, Feb. 2016.
[49] W. Wang, X. Yuan, X. Wu, and Y. Liu, "Fast image dehazing method based on linear transformation," *IEEE Trans. Multimedia*, vol. 19, no. 6, pp. 1142–1155, Jun. 2017.
[50] Y. Wang and C. Fan, "Single image defogging by multiscale depth fusion," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4826–4837, Nov. 2014.
[51] H. Wen, Y. Tian, T. Huang, and W. Gao, "Single underwater image enhancement with a new optical model," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2013, pp. 753–756.
[52] F. Wu *et al.*, "Weakly semi-supervised deep learning for multi-label image annotation," *IEEE Trans. Big Data*, vol. 1, no. 3, pp. 109–122, Sep. 2015.
[53] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 341–349.
[54] 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, Jan. 2014.
[55] Q. Yan, L. Xu, and J. Jia, "Dense scattering layer removal," in *Proc. SIGGRAPH Asia Tech. Briefs*, 2013, Paper 14.
[56] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision(Lecture Notes in Computer Science)*, vol. 8689. Berlin, Germany: Springer, 2014, pp. 818–833.
[57] Q. Zhu, J. Mai, and L. Shao, "A fast single image haze removal algorithm using color attenuation prior," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3522–3533, Nov. 2015.

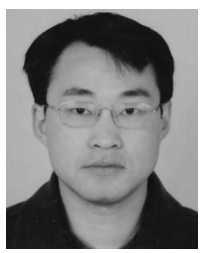


Yafei Song received the B.E. degree from the Beijing Institute of Technology, Beijing, China, in 2010, and the Ph.D. degree from Beihang University, Beijing, China, in 2017. He is currently a Researcher in the Samsung Research and Development Institute, Beijing, China. His research interests include computer vision and augmented reality.

Jia Li (M'12–SM'15) received the B.E. degree from Tsinghua University, Beijing, China, in 2005, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2011. He is currently an Associate Professor in the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing, China. His research interests include computer vision and image/video processing.



Xiaogang Wang is currently working toward the Ph.D. degree in the State Key Laboratory of Virtual Reality Technology and System, School of Computer Science and Engineering, Beihang University, Beijing, China. His research interest includes computer graphics.



Xiaowu Chen (M'09–SM'15) received the Ph.D. degree from Beihang University, Beijing, China, in 2001. He is currently a Professor in the State Key Laboratory of Virtual Reality Technology and Systems as well as the School of Computer Science and Engineering, Beihang University. His research interests include computer vision, computer graphics, virtual reality, and augmented reality.