Effects of Image Degradation and Degradation Removal to CNN-Based Image Classification

Yanting Pei, Yaping Huang, Qi Zou, Xingyuan Zhang, and Song Wang, *Senior Member, IEEE*

Abstract—Just like many other topics in computer vision, image classification has achieved significant progress recently by using deep learning neural networks, especially the Convolutional Neural Networks (CNNs). Most of the existing works focused on classifying very clear natural images, evidenced by the widely used image databases, such as Caltech-256, PASCAL VOCs, and ImageNet. However, in many real applications, the acquired images may contain certain degradations that lead to various kinds of blurring, noise, and distortions. One important and interesting problem is the effect of such degradations to the performance of CNN-based image classification and whether degradation removal helps CNN-based image classification. More specifically, we wonder whether image classification performance drops with each kind of degradation, whether this drop can be avoided by including degraded images into training, and whether existing computer vision algorithms that attempt to remove such degradations can help improve the image classification performance. In this article, we empirically study those problems for nine kinds of degraded images—hazy images, motion-blurred images, fish-eye images, underwater images, low resolution images, salt-and-peppered images, images with white Gaussian noise, Gaussian-blurred images, and out-of-focus images. We expect this article can draw more interests from the community to study the classification of degraded images.

Index Terms—Image classification, image degradation, degradation removal, CNN

1 Introduction

Associating an input image with one of the priorly specified image class, *image classification* is a fundamental and important problem in computer vision and artificial intelligence. While image classification [1], [2] has been studied in different applications for a long time, its performance is substantially improved in recent years by using supervised deep learning, e.g., Convolutional Neural Networks (CNNs) [3], [4], [5], [6], which unify the feature extraction and classification into a single end-to-end network. For example, on ImageNet dataset, a recent CNN-based image classification method [6] achieved a top-5 accuracy of 97.8 percent.

However, most of these excellent image classification performances are achieved on clear natural images, such as the images in databases of Caltech-256 [7], PASCAL VOCs [8] and ImageNet [9]. In many real applications, such as those related to autonomous driving, underwater robotics, video surveillance, wearable cameras, and medical imaging, the acquired images are not always clear. Instead, they suffer from various kinds of degradations. For example, images taken in the hazy weather, images taken by moving cameras, and images taken underwater by waterproof cameras usually contain different levels of intensity blurs. Images taken by fish-eye cameras usually show spatial distortions. Images taken by security and surveillance cameras, and medical imaging facilities, may produce low resolution images, salt-and-peppered images and Gaussian-blurred images. White Gaussian noise is very common in imaging since it models thermal chip noise and approximates the Poisson shot noise. Many images captured by smartphone cameras may contain different levels of out-of-focus blur. Some examples of degraded images are shown in Fig. 1. One important and interesting problem is whether the excellent classification performance obtained on clear natural images can be preserved on such degraded images by using the same deep learning techniques. Besides, does degradation removal, which has been long studied in the computer vision community, can help much image classification? In this paper, we empirically study these problems by constructing datasets of various kinds of degraded images, and further quantitatively evaluate and compare the CNN-based image classification models on these datasets.

More specifically, in this paper we select nine kinds of degraded images – hazy images, motion-blurred images, fish-eye images, underwater images, low resolution images, salt-and-peppered images, images with white Gaussian noise, Gaussian-blurred images and out-of-focus images – for our empirical study. To quantify the classification performance under different levels of degradations, we use their respective physical models to synthesize a large number of degraded images, as well as collect real hazy images from Internet. We then implement the CNN model using AlexNet [3], VGGNet [4] and ResNet [5] on Caffe and use them for classifying images with different degradation levels. The CNN-based model employs supervised learning

- *Y. Pei, Y. Huang, Q. Zou, and X. Zhang are with the School of Computer and Information Technology and Beijing Key Laboratory of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing 100044, China. E-mail: {yantingpei, yphuang, qzou, 15112071}@bjtu.edu.cn.*
- *S. Wang is with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29201 USA, and also with the College of Intelligence and Computing, Tianjin University, Tianjin 300072, China. E-mail: songwang@cec.sc.edu.*

Manuscript received 31 Jan. 2019; revised 22 Aug. 2019; accepted 24 Oct. 2019. Date of publication 1 Nov. 2019; date of current version 4 Mar. 2021. (Corresponding authors: Yaping Huang and Song Wang.)
Recommended for acceptance by P. Favaro.
Digital Object Identifier no. 10.1109/TPAMI.2019.2950923

Fig. 1. Examples of degraded images. From left to right and top to down are a hazy image, a motion-blurred image, a fish-eye image, an underwater image, a low resolution image, a salt-and-peppered image, an image with white Gaussian noise, a Gaussian-blurred image, and an out-of-focus image, respectively.

and requires a set of images for training. To more comprehensively explore the effect of degradations to image classification, we study not only the training and testing on the images with the same level of degradation, but also the training and testing on images with different levels of degradations. Besides, we also study whether image degradation removal helps CNN-based image classification.

The main contributions of this paper are threefolds. First, we conduct an empirical study of the effect of nine kinds of typical image degradations to the CNN-based image classification. Second, we use respective physical models to construct synthetic images with different levels of degradations for quantitative evaluation, and we also collect a real hazy image dataset from the Internet for the proposed empirical study. Third, we investigate whether existing degradation removal algorithms can benefit much the degraded image classification. Note that the goal of this paper is not the development of a new method to improve the classification performance on degraded images. Instead, we study whether the degraded image classification is a more challenging problem compared to the classification of clear natural images and whether its performance can be improved by selecting or pre-processing the training/test data. We expect this work can help attract more interests from the community to further study the image classification of degraded images, including the development of new methods for improving the image classification performance.

The remainder of the paper is organized as follows. Section 2 overviews the related work. Section 3 introduces the construction of degraded images, the collected real hazy image dataset and the evaluation metrics. Section 4 reports the datasets and effects of image degradation to image classification. Section 5 reports whether degradation removal helps image classification, followed by a brief conclusion in Section 6.

2 Related Work

In this section, we will briefly introduce the related works of image classification and the related works on various degraded images including hazy images, motion-blurred images, fish-eye images, underwater images, low resolution images, salt-and-peppered images, images with white Gaussian noise, Gaussian-blurred images and out-of-focus images.

2.1 Image Classification

Just like many other topics in computer vision, the performance of image classification has been significantly improved by using deep learning techniques, especially the CNN. In 2012, a CNN-based method [3] achieved a top-5 classification accuracy of 83.6 percent on ImageNet dataset in the ImageNet Large-Scale Visual Recognition Challenge 2012 (ILSVRC 2012). It is about 10 percent higher than the traditional methods [2] that achieved a top-5 accuracy of 74.3 percent on ImageNet dataset in ILSVRC 2011. Almost all the recent works and the state-of-the-art performance on image classification were achieved by CNN-based methods. For example, VGGNet in [4] increased the network depth using an architecture with very small convolution filters and achieved a top-1 accuracy of 75.2 percent and a top-5 accuracy of 92.5 percent on ImageNet dataset in ILSVRC 2014. Image classification accuracy in ILSVRC 2014 was then further improved [10] in 2015 by increasing the depth and width of the network. In [5], residual learning was applied to solve the gradient disappearance problem and achieved a top-5 accuracy of 96.4 percent on ImageNet dataset in ILSVRC 2015. In [6], an architectural unit was proposed based on the channel relationship, which adaptively recalibrates the channel-wise feature responses by explicitly modeling interdependencies between channels, resulting in a top-5 accuracy of 97.8 percent on ImageNet dataset in ILSVRC 2017. In [11], image regions for gaining spatial invariance are aligned and strongly localized features are learned, resulting in 95.0, 93.4, 94.4 and 84.0 percent classification accuracies on PASCAL VOC 2007, PASCAL VOC 2012, Scene-15 and MIT-67 datasets, respectively. Although these CNN-based methods have achieved excellent performance on image classification, most of them were only applied to the classification of clear natural images.

Degraded image-based recognition and classification have been studied in several recent works. In [12], only the influence of special degradations to face recognition was analyzed when using deep CNN-based approaches. Furthermore, in [12], the training data are always clear images. In [13], special degradations of low image resolution were studied in the applications of face identification, digit recognition and font recognition. In [14], a CNN-based method was proposed for improving the recognition performance of low-quality images and videos by using pre-training, data augmentation, and other strategies. In this paper, we conduct an empirical study to comprehensively study the effects of various degradations to the performance of CNN-based image classification and investigate whether the use of degraded image in training and a pre-processing of degradation removal are helpful for image classification.

2.2 Degraded Images

For *hazy images*, many models and algorithms were developed for removing the haze and restoring the original clear

image. Early methods for haze removal focused on developing hand-crafted features based on the statistics or prior of clear images [15], [16], [17], [18], [19], [20], such as dark channel prior [15], color attenuation prior [19], non-local prior [20]. To avoid hand-crafted priors, recent works [21], [22], [23], [24], [25], [26] automatically learned haze relevant features by CNN. Cai *et al.* [21] proposed an end-to-end CNN-based haze removal method and the layers of CNN architecture were specially designed to embody the established priors in image dehazing. Ren *et al.* [22] proposed a multi-scale deep neural network for haze removal, and the network consisted of a coarse-scale net for a holistic transmission map and a fine-scale net for local refinement. Li *et al.* [23] designed an end-to-end network based on a re-formulated atmospheric scattering model, instead of estimating the transmission matrix and the atmospheric light separately. Ren *et al.* [24] proposed an end-to-end trainable neural network consisting of an encoder and a decoder for image dehazing. Most of these haze removal models and algorithms aim at restoring the original clear image in terms of pixel-wise intensity difference, which may not reflect the perceptual similarity between the restored image and the underlying clear image. To address this issue, Li *et al.* [25] proposed a single image dehazing method via conditional generative adversarial network (GAN) with L_1 loss, adversarial loss and perceptual loss. Qu *et al.* [26] also proposed an enhanced pix2pix dehazing network by embedding a perceptual loss in a GAN.

The motion of the camera or the captured objects can lead to motion blur to the acquired images. Liu *et al.* [27] proposed a blurred image classification and analysis framework for detecting images containing blurred regions and recognizing the blur types in those regions. Kalalembang *et al.* [28] presented a method of detecting unwanted motion blur effects. Gast *et al.* [29] proposed a parametric object motion model by combining with a segmentation mask to exploit localized, non-uniform motion blur. Besides, effective and efficient deblurring of *motion-blurred images* has become an important research topic in the past decades [30], [31], [32], [33], [34], [35], [36]. Particularly, [30] is a classic book that provides a good coverage of different motion deblurring models and algorithms. Pan *et al.* [31] presented a blind image deblurring method based on the dark channel prior. Nah *et al.* [32] proposed a multi-scale convolutional neural network to restore sharp images in an end-to-end manner where blur was caused by various sources. Tao *et al.* [35] proposed a coarse-to-fine scheme to remove motion blur. Similar to the research on image dehazing, perceptual loss has been considered in many recent deep-learning based motion deblurring models [33], [34], [36]. Ramakrishnan *et al.* [33] proposed a deep filter based on GAN architecture integrated with global skip connection and dense architecture in order to tackle motion blur. Kupyn *et al.* [34] presented an end-to-end learning approach for removing motion blur based on conditional GAN. Lu *et al.* [36] presented an unsupervised domain-specific method for single-image deblurring based on disentangled representations.

Fish-eye images can provide wide-angle view of a scene, but introduce spatial distortions to the covered scene and objects. Kannala *et al.* [37] proposed a generic camera model for both the conventional and wide-angle lens cameras, as well as developed a calibration method for estimating the parameters of the model. Fu *et al.* [38] discussed how to explicitly employ the distortion cues to detect the forgery object in fish-eye images. Hughes *et al.* [39] proposed a method to estimate the intrinsic and extrinsic parameters of fish-eye cameras. Krams *et al.* [40] addressed the problem of people detection in top-view fish-eye imaging. Baek *et al.* [41] proposed a method for real-time detection, tracking, and classification of moving and stationary objects using multiple fish-eye images.

The *underwater images* taken by waterproof cameras, or other imaging facilities, are usually highly blurred and recognizing the objects from an underwater image is an important problem for both civil and military applications. Oliver *et al.* [42] proposed a traditional feature matching method using linear sparse coding for underwater images. Akkaynak *et al.* [43] introduced the space of attenuation coefficients that can be used for many underwater computer vision tasks. Wang *et al.* [44] proposed a method for feeble object detection of underwater images through logical stochastic resonance with delay loop. Möller *et al.* [45] proposed an active learning method for the classification of species in underwater images from a fixed observatory. Rajeev *et al.* [46] proposed a segmentation technique for underwater images based on K-means and local adaptive thresholding. Chen *et al.* [47] proposed an underwater object segmentation method based on optical features.

For the research on *low-resolution images*, Wang *et al.* [13] studied the problem of low-resolution image classification. Besides, by estimating the underlying high-resolution image from a low-resolution image, single image super-resolution (SISR) [48], [49], [50], [51], [52], [53], [54] has drawn much interest in recent years. Dong *et al.* [48] first proposed a deep learning method for SISR, which directly learns an end-to-end mapping between the low and high-resolution images. Tai *et al.* [49] proposed a very deep CNN model (up to 52 convolutional layers) to strive for deep yet concise networks for image super-resolution. Haris *et al.* [50] proposed a deep back-projection networks for SISR. Han *et al.* [51] explored the dual-state recurrent network for SISR. Wang *et al.* [52] recovered realistic texture in SISR by deep spatial feature transform. Zhang *et al.* [53] proposed a novel residual dense network to address SISR. Hui *et al.* [54] proposed a fast and accurate SISR via information distillation network.

For the research on *salt-and-peppered images*, Dong *et al.* [55] proposed a dual reweighted Lp-norm model with a more reasonable weighting rule and weaker powers for salt-and-pepper noise removal. Fu *et al.* [56] proposed a patch-based contour prior denoising approach for salt-and-pepper noise. Fu *et al.* [57] proposed a CNN-based non-local switching filter for salt-and-pepper noise removal. Fu *et al.* [58] proposed a salt-and-pepper denoising method based on the generative classification. For the research on *images with white Gaussian noise*, Kai *et al.* [59] trained a set of fast and effective CNN denoisers and integrated them into model-based optimization for solving inverse problems. Lefkimmiatis *et al.* [60] designed a new network architecture for learning discriminative image models that can efficiently tackle the problem of grayscale and color image denoising.

For the research on *Gaussian-blurred images*, Flusser *et al.* [61] introduced a new theory of invariants to model and removed Gaussian blur. Many deblurring methods have been developed for removing Gaussian blur [62], [63], [64]. Al-amri *et al.* [62] proposed to use four types of image deblurring techniques based on Wiener filter, Regularized filter, Lucy Richardson deconvolution algorithm and Blind deconvolution algorithm, respectively, to handle Gaussian blurs. Yair *et al.* [63] proposed a variable splitting method for solving the resulting optimization problem. Yu *et al.* [64] developed a toolbox consisting of small-scale convolutional networks of different complexities for different image restoration tasks, including the removal of Gaussian blur. For the research on *out-of-focus images*, many methods [65], [66] were developed for defocus blur detection (DBD) that aims to detect out-of-focus regions in an image. Zhao *et al.* [65] proposed a learning strategy by breaking DBD problem into multiple smaller defocus blur detectors and thus estimation errors could cancel out each other. Tang *et al.* [66] proposed a deep neural network which recurrently fuses and refines multi-scale deep features (DeFusionNet) for defocus blur detection.

Among the nine kinds of image degradations, motion blur, Gaussian blur and out-of-focus, introduce various blurs to the image. Several blur-invariant approaches have been developed for handling such blurred images in high-level vision tasks, e.g., image classification. They can be divided into two categories: moment-based methods [67], [68] and invariant-distance-based methods [69], [70], [71]. Flusser *et al.* first introduced blur invariance based on image moments in [67] and then derived the image features that are invariant with respect to blur in [68]. Zhang *et al.* [69] proposed a log-Fourier representation of signals and images to define blur-insensitive metrics between images. Gopalan *et al.* [70] proposed a blur-robust descriptor for face recognition. Zhang *et al.* [71] presented a Riemannian framework for analyzing signals and images to achieve invariance to different levels of Gaussian blur.

Different from the above mentioned works on degraded images, in this paper, we conduct a comprehensive empirical study to quantify the effects of these nine kinds of degradations to CNN-based image classification. Some of these prior works investigated the removal of degradations. Later in this paper, we will study whether the removal of degradations using these methods can help the CNN-based image classification or not.

3 Method

In this section, we first discuss the construction of the images with different levels of degradations. Then, we introduce the real hazy image dataset collected from the Internet. Finally, we introduce the evaluation metrics, as well as our training strategies.

3.1 Synthesis of Degraded Images

For large-scale CNN-based training and testing, we synthesize nine kinds of degraded images from clear images. Specially, we synthesize hazy images by following [72] and we can control the degradation levels of the synthesized hazy images by varying the atmospheric scattering coefficient β. We synthesize motion-blurred images by following [73] and the degradation level is controlled by varying the length of the blur kernel l. We synthesize fish-eye images by following [37] and the degradation level is controlled by varying the distortion exponent e. We synthesize underwater images by convolving the input image with the simplified version of Dolins PSF model [42], [74] and we can control the degradation levels of the synthesized underwater images by varying the optical depth τ and the single scattering albedo ω. We follow [75], [76] to construct low-resolution images by downsampling the high-resolution images using the bicubic kernel with downsampling factor f. We follow the similar procedure reported in [61] to add salt-and-pepper noise and Gaussian blur to a clear image, where we vary the noise density s of the salt-and-peppers and the standard deviation g of the Gaussian kernel to control the degradation level of salt-and-pepper noise and Gaussian blur, respectively. We follow [60] to construct white Gaussian noise and the image degradation levels are controlled by varying the noise variance v. We synthesize out-of-focus images by following [77] where the degradation level is controlled by varying the radius r of circular averaging filter. Fig. 2 shows the examples of synthesized images of different degradation kinds and levels.

3.2 Real Hazy Images

While we can construct synthetic degraded images by following well-acknowledged physical models, real degradation models can be much more complicated and a study on synthetic degraded image datasets may not completely reflect what we may encounter on real degraded images. To address this issue, we take hazy images as an example and collect a new dataset of hazy images from the Internet. This new dataset contains 4,610 images from 20 classes and we named it as *Haze-20*. These 20 image classes are bird (231), boat (236), bridge (233), building (251), bus (222), car (256), chair (213), cow (227), dog (244), horse (237), people (279), plane (235), sheep (204), sign (221), street-lamp (216), tower (230), traffic-light (206), train (207), tree (239) and truck (223), and the contents in the parenthesis are the number of images collected for each class. The number of images per class varies from 204 to 279. Some examples in Haze-20 are shown in Fig. 3.

One important experiment in our empirical study is to train on clear images and test on degraded images. This is not a problem for our synthetic data since their underlying degradation-free clear images are available, i.e., the original Caltech-256 data. For collected real images in Haze-20, we do not have their underlying clear images. To address this issue, we collect a new *HazeClear-20* image dataset from the Internet, which consists of haze-free images that fall in the same 20 classes as in Haze-20. HazeClear-20 consists of 3,000 images, with 150 images per class.

3.3 Evaluation Metrics

We will quantitatively evaluate the performance of image restoration by applying existing degradation removal methods and the performance of image classification. Other than visual examination, Peak Signal-to-Noise Ratio (PSNR) [78] and Structural Similarity (SSIM) [79] are widely used for evaluating the performance of image restoration when the ground-truth degradation-free image is available for each degraded image. For image classification, classification

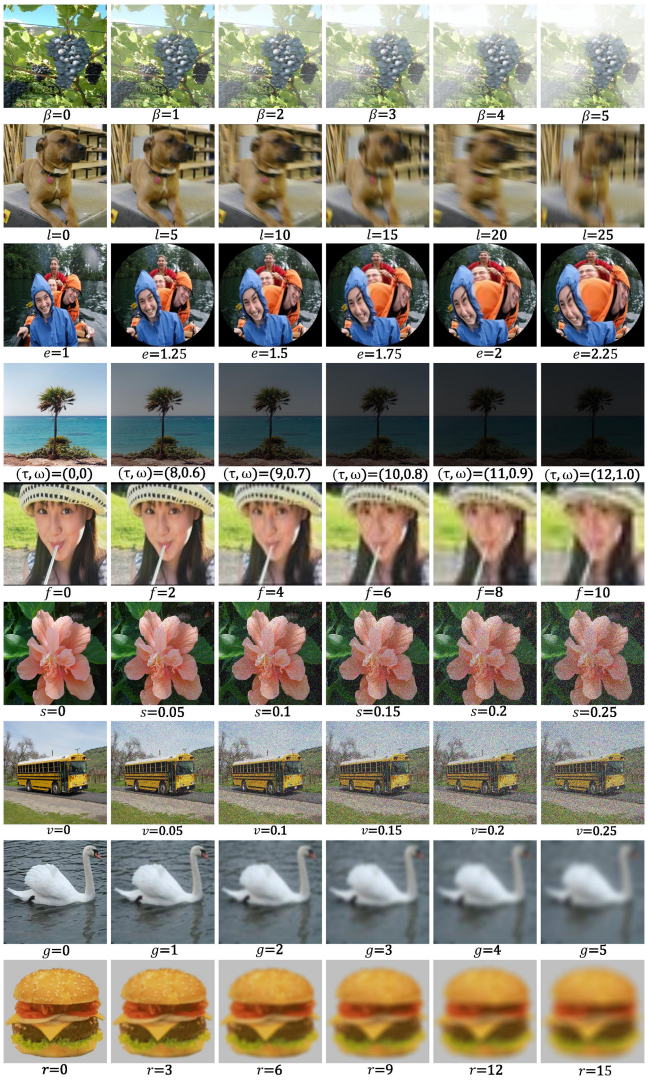


Fig. 2. Examples of the synthesized degraded images. From top to the bottom rows are hazy images, motion-blurred images, fish-eye images, underwater images, low resolution images, salt-and-peppered images, images with white Gaussian noise, Gaussian-blurred images, and out-of-focus images, respectively. The degradation levels are indicated below each image. The first column shows the original clear images.

accuracy $Accuracy=\frac{R}{N}$ is the most widely used performance evaluation metric, where N is the total number of testing images and R is the total number of testing images that are correctly classified by using the trained CNN-based models.

Note that, both PSNR and SSIM are objective metrics based on image statistics. Previous research has shown that they may not always be consistent with the image restoration quality perceived by human vision, which is quite subjective. In this paper, what we concern about is the performance of image classification after incorporating image restoration as preprocessing. Therefore, we will study whether PSNR and SSIM metrics show certain correlation to the image classification performance.

3.4 Training and Testing Methods

For a comprehensive analysis of the effects of image degradations to CNN-based image classification, we vary the selection of training and testing datasets for the CNN classifiers and design multiple experiments. In particular, for each kind of degradation: (1) we train and test CNN classifiers using image data of the different degradation levels; (2) we train and test CNN classifiers using image data of the same degradation level; (3) we train and test CNN classifiers using the images of the same type, including clear images, degraded images and restored images; (4) we train and test CNN classifiers on real hazy images to support our findings drawn from the study on the synthetic data, although the exact degradation levels of these real images are unknown; (5) we train CNN classifiers by combining images of different degradation levels and test on images of each degradation level; (6) we train CNN classifiers on clear images and test them using clear images, degraded images and restored images after applying different degradation-removal algorithms.

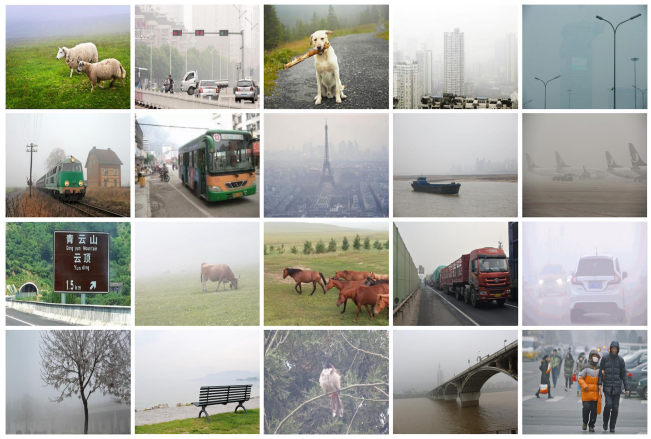


Fig. 3. Sample hazy images in our new Haze-20 dataset.

4 Effects of Image Degradation to CNN-Based Image Classification

In this section, we first describe the datasets and experiment settings. After that, we report the experiment results on synthetic and real image datasets. Finally, sample features extracted at each hidden layer are visualized to further analyze the experiment results.

4.1 Datasets and Experiment Setup

We synthesize degraded images using all the images in Caltech-256 dataset [7], which has been widely used for evaluating image classification algorithms. This dataset contains 30,607 images from 257 classes. For different kinds of degradation in each level, we synthesize 30,607 images by applying the respective model to each of the images in Caltech-256. In the original Caltech-256, we follow [4] to select 60 images randomly as training images per class, and the rest are used for testing. Among the training images, 20 percent per class are used as a validation set. We follow this strategy to split the synthesized image data accordingly: an image belongs to training set if it is synthesized from the original training data and belongs to testing set otherwise.

In our experiment, for each kind of degraded images, we select six different levels of degradations. More specifically, for hazy images, we set the parameter $\beta \in \{0, 1, 2, 3, 4, 5\}$. For motion-blurred images, we set the parameter $l \in \{0, 5, 10, 15, 20, 25\}$. For fish-eye images, we set the parameter

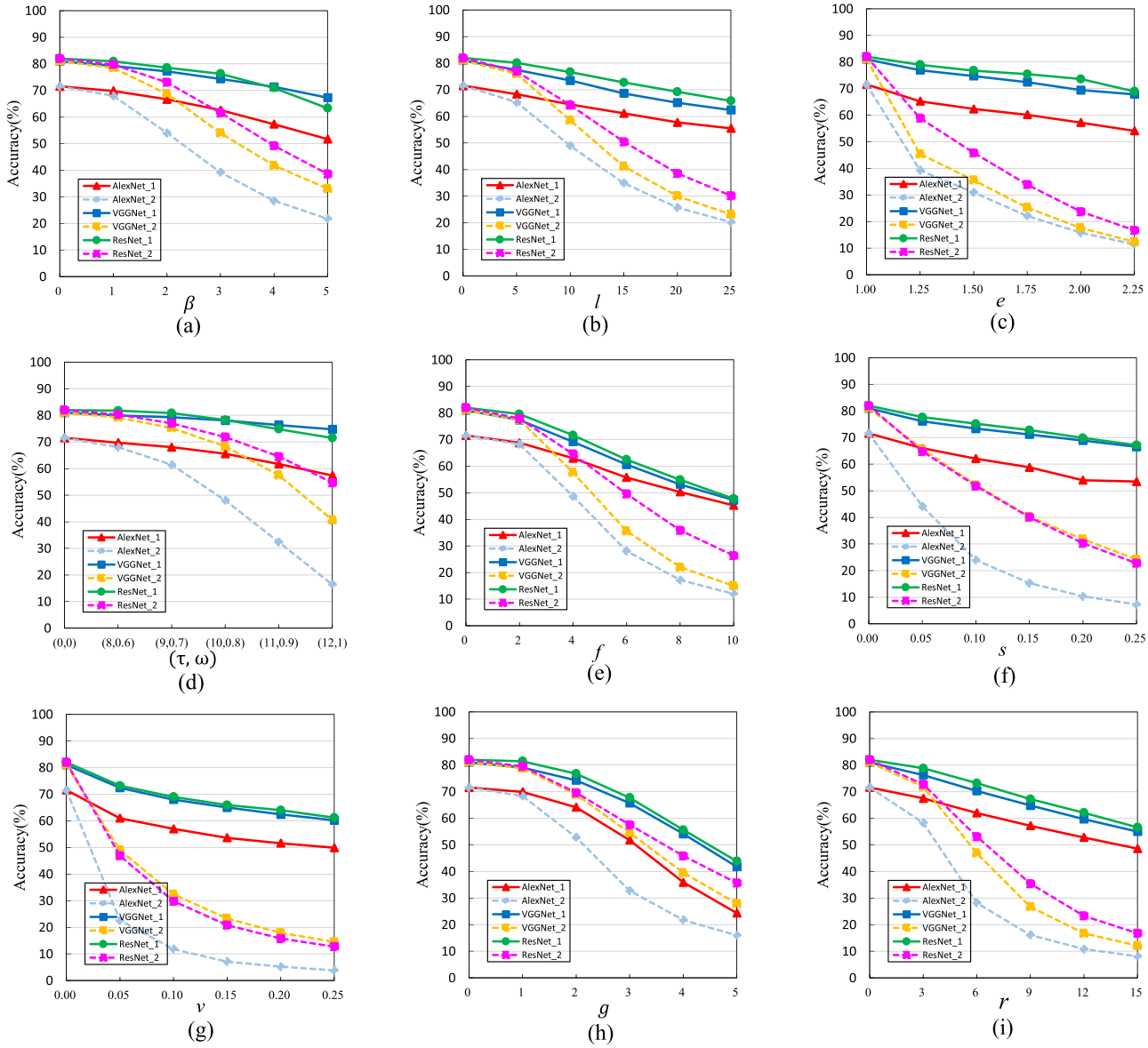


Fig. 4. The classification accuracy (%) of various degraded images synthesized from Caltech-256 dataset. From (a) to (i) are the accuracy (%) of hazy images, motion-blurred images, fish-eye images, underwater images, low resolution images, salt-and-peppered images, images with white Gaussian noise, Gaussian-blurred images, and out-of-focus images, respectively.

$e \in \{1, 1.25, 1.5, 1.75, 2, 2.25\}$. For underwater images, we set parameters $(\tau, \omega) \in \{(0,0), (8,0.6), (9,0.7), (10,0.8), (11,0.9), (12,1)\}$. For low resolution images, we set the parameter $f \in \{0, 2, 4, 6, 8, 10\}$. For salt-and-peppered images, we set the parameter $s \in \{0, 0.05, 0.1, 0.15, 0.2, 0.25\}$. For images with white Gaussian noise, we set the parameter $v \in \{0, 0.05, 0.1, 0.15, 0.2, 0.25\}$. For Gaussian-blurred images, we set the parameter $g \in \{0, 1, 2, 3, 4, 5\}$. For out-of-focus images, we set the parameter $r \in \{0, 3, 6, 9, 12, 15\}$.

For the collected real hazy images in Haze-20 dataset and clear images in HazeClear-20 dataset, we select 100 images randomly from each class as training images, and the rest are used for testing. Among the training images, 20 percent per class are used as a validation set.

We implement AlexNet [3], VGGNet-16 [4] and ResNet-50 [5] on Caffe in this paper. The CNN architectures are pre-trained on ImageNet dataset [9] that consists of 1,000 classes with 1.2 million training images. The pre-trained model is then fine tuned on our training images before the model is used for image classification. Note that even though we use AlexNet, VGGNet-16 and ResNet-50 for their simplicity, it is feasible to use other neural networks as well, such as SE-Net [6].

4.2 Results on Synthetic Images – Individual Degradation Levels

In order to verify the effect of the image degradation to CNN-based image classification, we train and test AlexNet, VGGNet and ResNet on the nine kinds of synthetic degraded images and the classification results are shown in Figs. 4a, 4b, 4c, 4d, 4e, 4f, 4g, 4h, 4i. *AlexNet_1*, *VGGNet_1* and *ResNet_1* indicate the *Case 1* where training and testing are conducted on the same degradation-kind and degradation-level images using AlexNet, VGGNet and ResNet, respectively. *AlexNet_2*, *VGGNet_2* and *ResNet_2* indicate the *Case 2* of training on clear images and testing on different level of synthesized

TABLE 1
The Classification Accuracy (%) of Hazy Images, Motion-Blurred Images, Fish-Eye Images, and Underwater Images by Using AlexNet and VGGNet-16 With Different Levels of Degradation, Respectively

β	0	1	2	3	4	5	l	0	5	10	15	20	25
0	**71.7**	67.9	54.0	39.3	28.5	21.8	0	**71.7**	65.3	48.9	34.9	25.7	20.2
1	68.7	**69.8**	65.4	55.2	43.4	33.5	5	68.4	**68.9**	62.6	52.8	42.3	33.3
2	67.2	68.4	**66.7**	61.9	54.0	44.2	10	64.2	66.6	**65.3**	61.4	56.1	49.4
3	64.7	66.3	65.5	**62.6**	57.3	50.3	15	58.9	62.9	63.4	**62.0**	59.0	55.2
4	61.3	63.5	63.0	60.9	**57.3**	**52.4**	20	53.7	58.9	60.5	60.5	**59.0**	**56.4**
5	58.8	61.1	61.0	59.0	55.9	51.7	25	49.5	55.7	58.6	58.7	57.7	56.2
e	1	1.25	1.5	1.75	2	2.25	(τ, ω)	(0,0)	(8,0.6)	(9,0.7)	(10,0.8)	(11,0.9)	(12,1)
1	**71.7**	39.2	31.0	22.1	15.8	11.4	(0,0)	**71.7**	68.0	61.5	48.1	32.3	16.4
1.25	65.6	**65.2**	61.4	53.7	42.8	31.2	(8,0.6)	68.9	**69.8**	65.9	57.9	44.2	26.6
1.5	61.4	63.5	**62.3**	58.7	52.1	43.2	(9,0.7)	64.8	67.7	**68.1**	64.3	57.9	45.0
1.75	56.2	60.1	61.3	**60.2**	56.5	51.3	(10,0.8)	58.6	64.5	66.7	**65.6**	**62.1**	55.3
2	51.7	56.7	58.4	58.4	**57.2**	54.0	(11,0.9)	52.5	59.9	64.1	63.9	61.8	57.3
2.25	46.4	52.9	55.5	55.8	54.4	**54.1**	(12,1)	47.2	54.2	59.5	60.8	60.4	**57.4**
β	0	1	2	3	4	5	l	0	5	10	15	20	25
0	**81.0**	78.5	68.7	54.0	41.8	33.0	0	**81.0**	76.0	58.5	41.2	30.0	23.1
1	79.4	**79.2**	75.9	68.2	57.5	47.2	5	78.0	**77.9**	71.3	59.3	45.8	35.4
2	77.9	78.4	**77.2**	74.3	67.7	58.8	10	74.4	75.9	**73.4**	69.0	66.2	61.7
3	76.6	77.2	76.4	**74.4**	70.4	64.3	15	70.8	73.3	72.4	**70.0**	66.2	61.7
4	74.5	75.5	75.4	74.0	**71.4**	66.8	20	65.3	69.0	70.4	68.8	**66.3**	**63.5**
5	71.7	73.2	73.8	73.1	70.7	**67.8**	25	62.5	66.4	67.7	66.9	65.4	62.9
e	1	1.25	1.5	1.75	2	2.25	(τ, ω)	(0,0)	(8,0.6)	(9,0.7)	(10,0.8)	(11,0.9)	(12,1)
1	**81.0**	45.4	35.5	25.3	17.6	12.4	(0,0)	**81.0**	79.3	75.3	68.3	57.5	40.7
1.25	75.9	**76.9**	72.2	64.0	53.4	41.5	(8,0.6)	79.5	**80.0**	78.4	75.8	70.8	63.0
1.5	73.7	76.3	**74.7**	70.3	63.6	55.6	(9,0.7)	77.8	78.7	**79.3**	77.7	74.6	69.1
1.75	70.1	74.6	74.4	**72.4**	69.0	63.9	(10,0.8)	76.5	77.8	78.6	**78.1**	75.9	72.0
2	68.6	72.3	73.1	71.8	69.4	66.1	(11,0.9)	74.5	75.7	77.6	77.4	76.4	73.6
2.25	63.7	70.3	71.7	71.4	**69.9**	**67.8**	(12,1)	73.1	74.2	76.4	76.9	**76.6**	**74.8**

degraded images using AlexNet, VGGNet and ResNet, respectively. We can see that with the increase of the degradation level of the test images, the classification accuracy decreases rapidly in Case 2. In Case 1, the classification accuracy is better than that in the corresponding level in Case 2. However, even if we train and test the CNN using the images of the same degradation kind and level, the classification performance still drops when the degradation level increases. This may be caused by the partial loss of discriminative image information in the image degradations.

We select four kinds synthesized degraded images – hazy images, motion-blurred images, fish-eye images and underwater images – for more comprehensive experiments. For each kind of them, we exhaustively try the training using the training set of one level and testing on the testing data of the same or another level. The image classification accuracies by using AlexNet and VGGNet-16 are summarized in the eight subtables in Table 1. The top four subtables show the classification accuracies of hazy images, motion-blurred images, fish-eye images and underwater images by using AlexNet, and the bottom four subtables show the classification accuracies of these four kinds of degraded images by using VGGNet-16. Rows indicate the kind of training images and the columns indicate the kind of test images. In these subtables, the accuracies along the diagonal are achieved by training and testing on the image data of the same degradation level. The accuracies at non-diagonal elements are achieved by training and testing on the image data of the different degradation levels. In these eight subtables, we highlight the maximum accuracy in each column.

From the results reported in these subtables, we can see that, the maximum values in each column are usually located along the diagonal of each subtable. This indicates that, to achieve the best possible accuracy in classifying the images with certain level of degradations, we need to collect the training images with the same kind of degradation and with the same or similar degradation levels. If the degradation levels of the training images and testing images have large gaps, the testing accuracy can be very low. For example, if we use clear images ($\beta = 0$) to train CNN classifiers, and then apply them to classify images with hazy level of $\beta = 5$, the accuracy will drop significantly from 81.0 percent (training and testing both on clear images) or 67.8 percent (training and testing both on $\beta = 5$ hazy images) to 33.0 percent. We also find that, in general, VGGNet-16 produces higher classification accuracy than AlexNet when using the same training set. This is not surprising since VGGNet-16 is a deeper network. We can find that the accuracy drop when using VGGNet-16 is usually less than the drop when using AlexNet. From this perspective, VGGNet-16 also outperforms AlexNet.

4.3 Results on Synthetic Images – Mixed Degradation Levels

In practice, we may not know exactly the degradation levels of real images, and it can be difficult to guarantee that the degradation levels of the testing images match those of the

TABLE 2
Classification Accuracy (%) When VGGNet-16 CNN Classifiers are Trained by Mixing Different-Level Training Images of the Same Degradation Kind

hazy images:						
β	0	1	2	3	4	5
Accuracy	79.7	79.5	78.4	76.7	73.9	69.5
motion-blurred images:						
l	0	5	10	15	20	25
Accuracy	77.9	76.7	74.5	71.8	68.9	66.4
fish-eye images:						
e	1	1.25	1.5	1.75	2	2.25
Accuracy	78.7	77.6	76.5	74.7	73.1	70.4
underwater images:						
(τ, ω)	(0,0)	(8,0.6)	(9,0.7)	(10,0.8)	(11,0.9)	(12,1.0)
Accuracy	80.2	79.9	79.7	79.2	77.9	76.0
low resolution images:						
f	0	2	4	6	8	10
Accuracy	76.2	73.9	68.4	62.4	56.4	51.0
salt-and-peppered images:						
s	0	0.05	0.1	0.15	0.2	0.25
Accuracy	76.9	75.5	73.9	72.7	71.2	69.5
images with white Gaussian noise:						
v	0	0.05	0.1	0.15	0.2	0.25
Accuracy	74.8	71.4	69.3	67.4	66.1	64.3
Gaussian-blurred images:						
g	0	1	2	3	4	5
Accuracy	78.8	78.0	75.5	73.0	70.0	67.0
out-of-focus images:						
r	0	3	6	9	12	15
Accuracy	77.0	74.6	70.8	67.0	63.5	59.4

training images. Therefore, it is important to study the case where training images are mixed with a wide range of image degradation levels.

For each kind of degradation, we combine the training images of all different degradation levels to generate a mixed training set for CNN training. Then, we test the CNN classifiers on testing images of the same degradation kind at each degradation level. Results are shown in Table 2, and the nine subtables from top to bottom are classification accuracies (%) of hazy images, motion-blurred images, fish-eye images, underwater images, low resolution images, salt-and-peppered images, images with white Gaussian noise, Gaussian images and out-of-focus images by using VGGNet-16, respectively.

From Table 2, we can see that the classification accuracies of clear images in its nine subtables are 79.7, 77.9, 78.7, 80.2, 76.2, 76.9, 74.8, 78.8 and 77.0 percent respectively, which are lower than the clear-image classification accuracy of 81 percent shown in Table 1, where training images only contain clear images. This indicates that the inclusion of degraded images into training may affect the classification accuracy of clear images. However, we can also see that the accuracy of each degradation kind and each degradation level in Table 2 is usually higher than the accuracy of the corresponding degradation kind and level shown along the diagonal of the bottom four subtables in Table 1, except for the degradation-free clear images. This indicates that, if we know that the test images are degraded, we may want to include as many degraded images as possible, even of different degradation levels, into the training set.

TABLE 3
Classification Accuracy (%) on Haze-20 and HazeClear-20 Datasets Using VGGNet and AlexNet, Respectively

	AlexNet		VGGNet	
	HazeClear-20	Haze-20	HazeClear-20	Haze-20
HazeClear-20	97.2	49.9	98.0	62.3
Haze-20	82.0	75.4	88.7	81.2
Combine	96.7	75.9	97.9	81.4

4.4 Results on Real Hazy Images

We conduct experiments in Haze-20 and HazeClear-20 datasets using AlexNet and VGGNet-16, respectively. The experimental results are shown in Table 3. Rows indicate the kind of training images and the columns indicate the kind of testing images. In particular, "Combine" indicates the combination of the haze and clear training images for training. We can see that when we train and test on clear images, the accuracy can be further improved to 98.0 percent using VGGNet-16. However, when we train and test on real hazy images, the accuracy drops to only 81.2 percent using VGGNet-16. When the training set mixes hazy and clear images, the test accuracy on clear images and hazy images is 97.9 and 81.4 percent, respectively. This trend is largely consistent with what we find from the above synthetic image experiments.

4.5 Hidden-Layer Features

We can scrutinize the features extracted at each hidden layer to analyze the possible reasons that cause the performance drop in degraded image classification. We extract features of the first to the fifth max-pooling layers in VGGNet-16, and they are labeled as $pool_1$, $pool_2$, $pool_3$, $pool_4$, and $pool_5$ respectively in Fig. 5. For better visual effects, we resize those features using the bi-cubic interpolation, such that they have the same size as the input image.

We can see that, compared to clear images, the activations of "$pool_1$" and "$pool_2$" of the degraded images are not particularly discriminative. As we all know, shallow layers of CNN

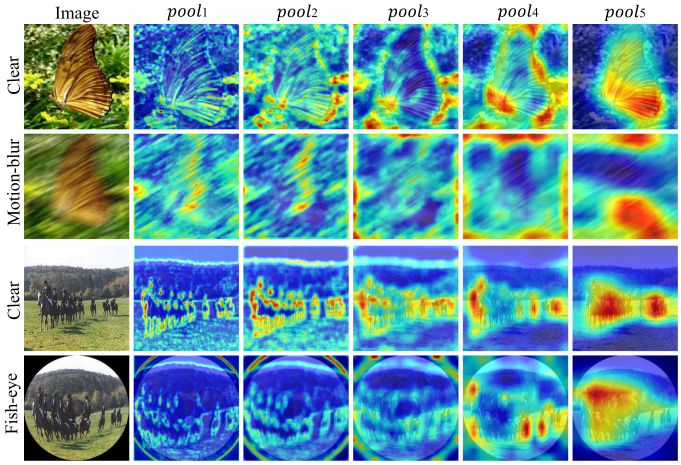


Fig. 5. Activations of hidden layers of CNN on image classification. From left to right are input images, and the activations at $pool_1$, $pool_2$, $pool_3$, $pool_4$, and $pool_5$ layers, respectively.

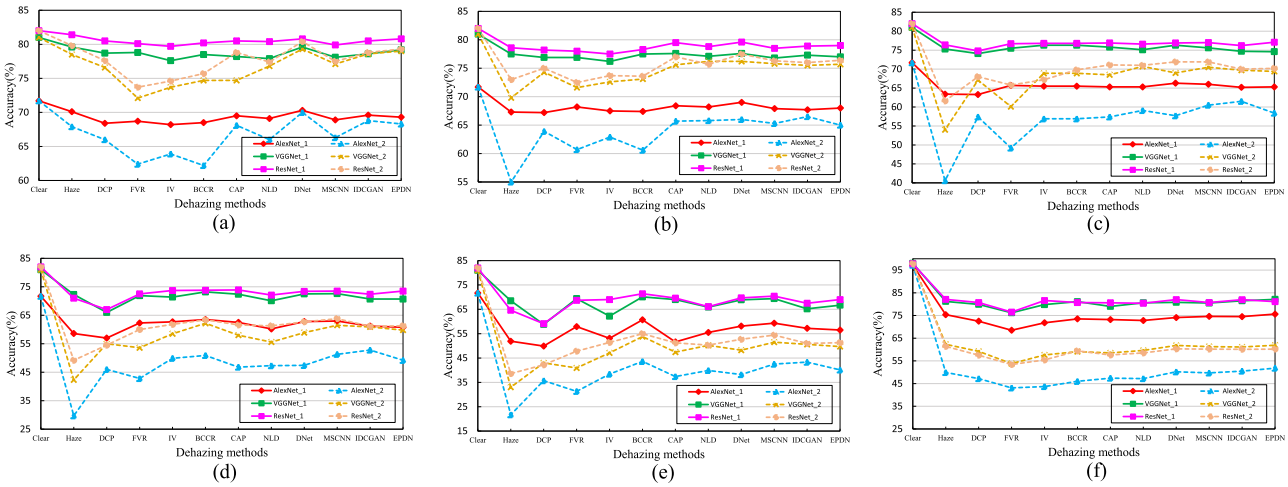


Fig. 6. The classification accuracy on different hazy images. (a)-(e) Classification accuracies on testing synthetic hazy images with $\beta = 1, 2, 3, 4, 5$, respectively. (f) Classification accuracy on the testing images in Haze-20.

reflect color, texture, and other low-level image features. The low quality of these low-level features can very negatively affect the extraction of good-quality high-level features in high layers, leading to decreased image-classification accuracies. Besides, from the high-level features of degraded image, we can see that the salient region is not accurately localized, e.g., the butterfly in the motion-blurred image in Fig. 5. The main reason is that distortion changes the shape and appearance of objects, leading to incorrect features in CNN layers and final errors in image classification, e.g., the incorrect recognition of horse in the fish-eye image in Fig. 5.

5 Does Degradation Removal Help Image Classification?

As discussed earlier, for many kinds of image degradations, many researches have been conducted to remove/reduce the image degradation with the goal of restoring the underlying clear images. One interesting problem is whether we can get better classification accuracy by training and testing on the restored images after the degradation removal. In this section, we take haze removal and motion-blur removal as examples to study whether degradation removal can help image classification.

5.1 Does Haze Removal Help Image Classification?

In this study we try ten state-of-the-art image-dehazing methods: Dark-Channel Prior (DCP) [15], Fast Visibility Restoration (FVR) [16], Improved Visibility(IV) [17], Boundary Constraint and Contextual Regularization (BCCR) [18], Color Attenuation Prior (CAP) [19], Non-local Image Dehazing (NLD) [20], DehazeNet (DNet) [21], MSCNN [22], IDCGAN [25] and EPDN [26]. We examine each of them to see whether it can help improve the performance of hazy image classification.

5.1.1 *Quantitative Comparisons on Synthetic and Real Hazy Images*

To verify whether haze-removal preprocessing can improve the performance of hazy image classification, we perform quantitative evaluation on the synthetic and real hazy images with and without haze removal. The classification results are shown in Fig. 6, where (a-e) are the classification accuracies on testing synthetic hazy images with $\beta = 1, 2, 3, 4, 5$, respectively using different dehazing methods. For these five curve figures, the horizontal axis represents different dehazing methods, where "Clear" indicates the use of the testing images in the original Caltech-256 dataset and this assumes a perfect image dehazing in the ideal case. The case of "Haze" indicates the testing on the hazy images without any dehazing. (f) is the classification accuracy on the testing images in Haze-20 using different dehazing methods, where "Clear" indicates the use of testing images in HazeClear-20 and "Haze" indicates the use of testing images in Haze-20 without any dehazing. *AlexNet_1*, *VGGNet_1* and *ResNet_1* represent the case of training and testing on the same kinds of images, e.g., training on the training images in Haze-20 after DCP dehazing, then testing on testing images in Haze-20 after DCP dehazing, by using AlexNet, VGGNet and ResNet, respectively. *AlexNet_2*, *VGGNet_2* and *ResNet_2* represent the case of training on clear images, i.e., for (a-e), we train on training images in original Caltech-256, and for (f), we train on training images in HazeClear-20, by using AlexNet, VGGNet and ResNet, respectively.

We can see that when we train CNN models on clear images and test them on hazy images with and without haze removal (e.g., *AlexNet_2*, *VGGNet_2* and *ResNet_2*), the classification performance drop significantly. From Fig. 6e, image classification accuracy drop from 71.7 to 21.7 percent when images have a haze level of $\beta = 5$ by using AlexNet. Along the same curve shown in Fig. 6e, we can see that by applying a dehazing method on the testing images, the classification accuracy can move up to 42.5 percent (using MSCNN dehazing). But it is still much lower than 71.7 percent, the accuracy on classifying original clear images. These experiments indicate that haze significantly affects the accuracy of CNN-based image classification when training on original clear images, even if the test image is preprocessed by a dehazing algorithm. However, if we directly train the classifiers on the hazy image of the same level, the classification accuracy moves up to 51.9 percent, as shown in the red curve in Fig. 6e, where no dehazing is involved in training and testing images. Another choice is to apply the same dehazing methods to both training and testing images.

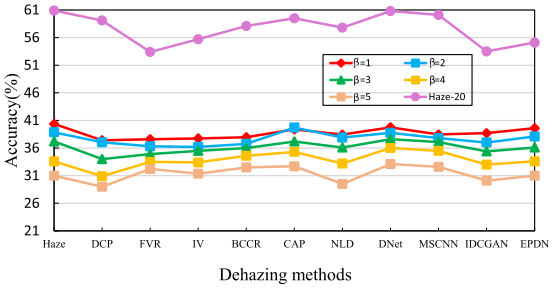


Fig. 7. Classification accuracy (%) on synthetic and real-world hazy images by using a non-CNN-based image classification method. Here the same kinds of images are used for training, i.e., building the basis for sparse coding, and testing, just like the case corresponding to the solid curves (*AlexNet_1*, *VGGNet_1*, and *ResNet_1*) in Fig. 6.

From results shown in all the six subfigures in Fig. 6, we can see that the resulting accuracy is similar to the case where no dehazing is applied to training and testing images. This indicates that the dehazing conducted in this study does not help the CNN-based image classification much. We believe this is due to the fact that the dehazing does not introduce new information to the image.

There are also many non-CNN-based image classification methods. While it is difficult to include all of them into our empirical study, we try the one based on sparse coding [1] and the results are shown in Fig. 7, where $\beta = 1, 2, 3, 4, 5$ represent haze levels of synthetic hazy images constructed on Caltech-256 dataset and *Haze-20* represents Haze-20 dataset. For this specific non-CNN-based image classification method, we can get the similar conclusion that the tried dehazing does not help image classification much. Comparing Figs. 6 and 7, we can see that the classification accuracy of this non-CNN-based method is much lower than the state-of-the-art CNN-based methods. Therefore, we focus on CNN-based image classification in this paper.

In Figs. 6 and 7, IDCGAN and EPDN are two dehazing methods with perceptual loss while other dehazing methods do not include perceptual loss. We can see that the IDCGAN and EPDN do not lead to better image classification than the other dehazing methods. For example, in Fig. 6a, the classification accuracies with IDCGAN/EPDN dehazing using AlexNet (the red solid curve), VGGNet (the green solid curve) and ResNet (the purple solid curve) are 69.6%/69.3%, 78.6%/79.2%, 80.5%/80.9%, respectively. The highest accuracies with other dehazing methods using AlexNet, VGGNet and ResNet are 70.3 percent (DNet), 79.6 percent (DNet), 80.8 percent (DNet), respectively.

5.1.2 Training on Mixed-Level Hazy Images

For more comprehensive analysis of dehazing methods, we conduct experiments of training on hazy images with mixed haze levels. For synthetic dataset, we try two cases. In Case 1, we mix all six levels of hazy images by selecting 10 images per class from each level of hazy images as training set and among the training images, two images per class per haze level are taken as validation set. We then test on the testing images of the involved haze levels – actually all six levels for this case – respectively. Results are shown in Fig. 8a, 8b and 8c when using AlexNet, VGGNet and ResNet respectively. In Case 2, we randomly choose images from two different haze levels and mix them. In this case, 30 images per class per level are taken as training images and among the training images, 6 images per class per level are used as validation images. This way we have 60 images per class for training. Similarly, we then test on the testing images of the involved two haze levels, respectively. Results are shown in Fig. 8d and 8e for four different kinds of level combinations, respectively. For real hazy images, we mix clear images in HazeClear-20 and hazy images in Haze-20 by picking 50 images per class for training and then test on the testing images in Haze-20 and HazeClear-20 respectively. Results are shown in Fig. 8f. Similarly, combining all the results, the use of dehazing does not clearly improve the image classification accuracy, over the case of directly training and testing on hazy images.

5.1.3 Performance Evaluation of Dehazing Methods

In this section, we study whether there is a correlation between the dehazing metrics PSNR/SSIM and the image

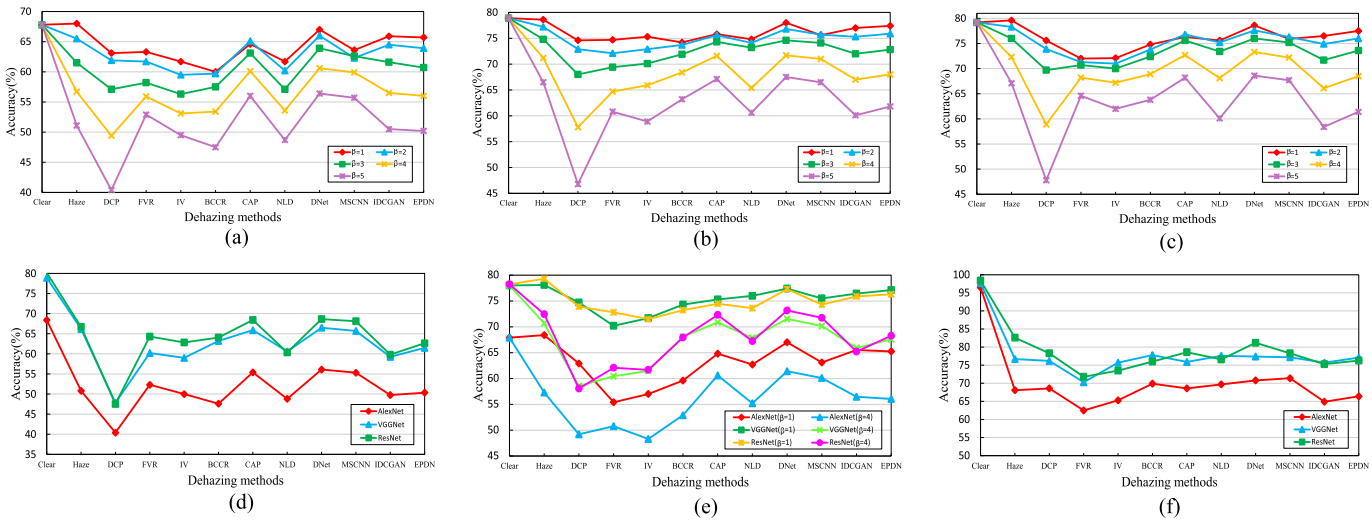


Fig. 8. Classification accuracy when training on mixed-level hazy images. (a), (b), and (c) Mix all six levels of synthetic images. (d) Mix two levels $\beta = 0$ and $\beta = 5$. (e) Mix two levels $\beta = 1$ and $\beta = 4$. (f) Mix Haze-20 and HazeClear-20.

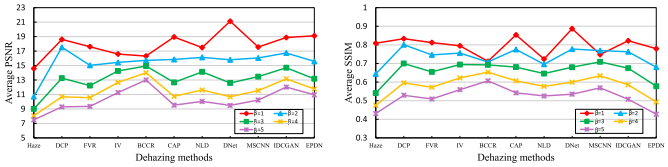


Fig. 9. Average PSNR and SSIM values on synthetic image dataset at different haze levels.

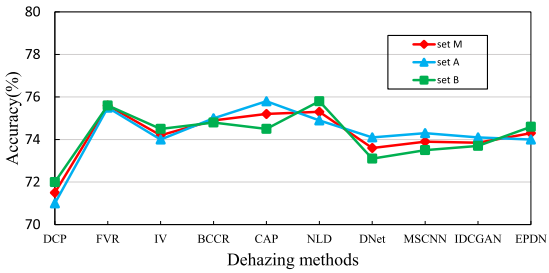


Fig. 10. Classification accuracy of different sets of dehazed images subjectively selected by human.

classification performance. On the synthetic images, we can compute the metrics PSNR and SSIM on all the dehazing results, which are shown in Fig. 9. In this figure, the PSNR and SSIM values are averaged over the respective testing images. We pick the red curves (*AlexNet_1*) from Fig. 6a, 6b, 6c, 6d, 6e and for each haze level in $\beta = 1, 2, 3, 4, 5$, we rank all the dehazing methods based on the classification accuracy. We then rank these methods based on average PSNR and SSIM at the same haze level. Finally we calculate the rank correlation between image classification and PSNR/SSIM at each haze level. Results are shown in Table 4. Negative values indicate negative correlation, positive values indicate positive correlation and the greater the absolute value, the higher the correlation. We can see that their correlations are actually low.

5.1.4 Subjective Evaluation

In this section, we conduct an experiment for subjective evaluation of the image dehazing. By observing the dehazed images, we randomly select 10 images per class with $\beta = 3$ and subjectively divide them into 5 with better dehazing effect and 5 with worse dehazing effect. This way, we have 2,570 images in total (set M) and 1,285 images each with better dehazing (set A) and worse dehazing (set B). Classification accuracy (%) using VGGNet is shown in Fig. 10 and we can see that there is no significant accuracy difference for these three sets. This indicates that the classification accuracy is not consistent with the human subjective evaluation of the image dehazing quality.

5.1.5 Feature Reconstruction

The CNN networks used for image classification consist of multiple layers to extract deep image features. One interesting question is whether certain layers in the trained CNN actually perform image dehazing implicitly. We pick a reconstruction method [80] to reconstruct the image according to feature maps of all the layers in AlexNet. The reconstruction results are shown in Fig. 11, from which we can see that, for the first several layers, the reconstructed images do not show any dehazing effect. For the last several layers, the reconstructed images have been distorted, let alone dehazing. One possibility of this is that many existing image dehazing methods aim to please human vision system, which may not benefit CNN-based image classification. Meanwhile, many existing image dehazing methods introduce information loss, such as color distortion, and may increase the difficulty of image classification.

TABLE 4
The Rank Correlation Between Image-Classification Accuracy and PSNR/SSIM at Each Haze Level

Correlation	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
(Accuracy, PSNR)	0.3455	-0.0545	0.4061	0.0667	0.2485
(Accuracy, SSIM)	-0.0909	-0.2970	-0.3333	0.1636	-0.2606

5.2 Does Motion-Blur Removal Help Image Classification?

We try five state-of-the-art motion deblur methods: D-DCP [31], DeepDeblur [32], DGF [33], DeblurGAN [34] and SRN [35]. We examine each of them to see whether it can help improve the performance of motion-blurred image classification.

5.2.1 Quantitative Comparisons on Motion-blurred Images

To verify whether motion-blur removal pre-processing can improve the performance of motion-blurred image classification, we test on synthetic motion-blurred images with and without motion-blur removal for quantitative evaluation. The classification results are shown in Fig. 12, where (a-e) show the classification accuracies on testing synthetic motion-blurred images with $l = 5, 10, 15, 20, 25$, respectively using different deblurring methods. Six curves in this figure correspond to those in Fig. 6 except that motion deblurring is considered here instead of dehazing. We can see that the accuracy is similar to the case where no deblurring is applied to training and testing images. This indicates that, just like dehazing, the motion deblurring preprocessing can not help image classification much. In Fig. 12, DGF and DeblurGAN are motion-blur removal methods with perceptual loss, while D-DCP, DeepDeblur and SRN do not consider perceptual loss. As for dehazing methods, the consideration of perceptual loss in motion-deblurring does not lead to better classification of motion-blurred images.

5.2.2 Performance Evaluation of Deblurring Methods

In this section, we study whether there is a correlation between the deblurring metrics PSNR/SSIM and the image classification performance. On the synthetic images, we can compute the metrics PSNR and SSIM on all the deblurring results, which are shown in Fig. 13. In this figure, the PSNR and SSIM values are averaged over the respective testing images. We pick the red curves (*AlexNet_1*) from Fig. 12 and for each motion-blur level in $l = 5, 10, 15, 20, 25$, we rank all the deblurring methods based on the classification accuracy. We then rank these methods based on average PSNR and

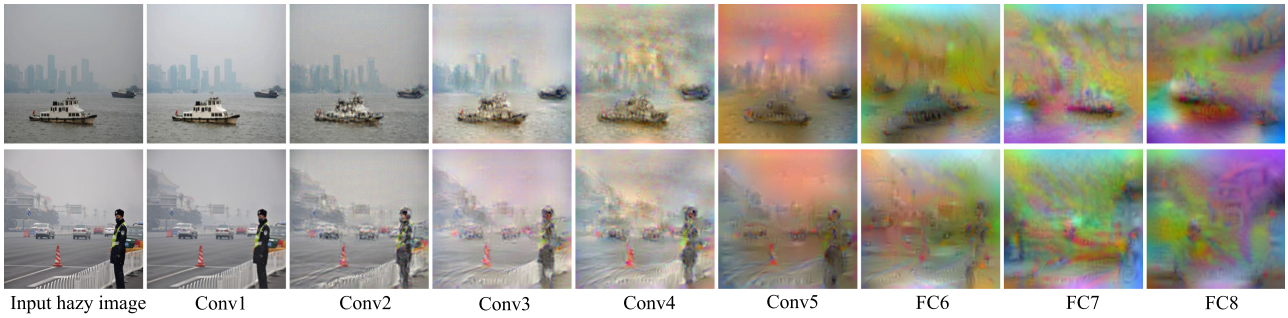


Fig. 11. Reconstruction results for two images, shown in two rows, respectively. The leftmost column shows the input hazy images and the following columns are the images reconstructed from different layers in AlexNet.

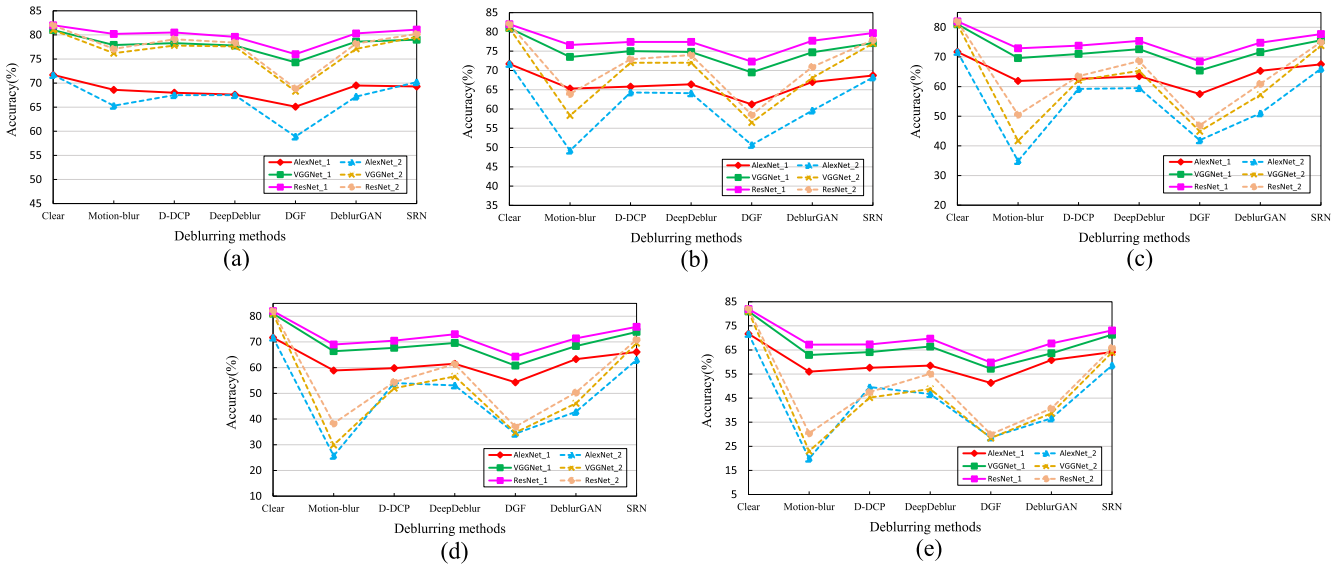


Fig. 12. The classification accuracy on different motion-blurred images. (a-e) Classification accuracies on testing synthetic motion-blurred images with $l = 5, 10, 15, 20, 25$, respectively.

SSIM at the same motion-blur level. Finally we calculate the rank correlation between image classification and PSNR/SSIM at each motion-blur level. Results are shown in Table 5. We can see that their correlations are usually low.

5.3 Comparison to Blur-Invariant Approaches

Many blur-invariant approaches, including moment-based methods [67], [68] and invariant-distance-based methods [69], [70], [71] were also developed for handling images with various kinds of blurs. In this section, we include some of them into performance evaluation.

Specifically, we consider three blur-invariant approaches: the moment-invariants (MI) [68], the blur-invariant metric based on the log-Fourier transform (LFT metric) [69], and the blur-invariant metric (BI metric) [71], in this section.

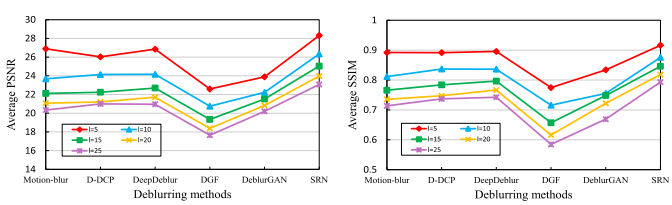


Fig. 13. Average PSNR and SSIM values on synthetic image dataset at different motion-blur levels.

Following [71], we combine MI, LFT metric and BI metric into the nearest-neighbor classifiers for image classification. We randomly select 10 image classes, including backpack, beer-mug, butterfly, dog, grapes, laptop, people, school-bus, speed-boat, tennis-shoes, in Caltech-256 dataset, and there are 480 training images and 698 testing images. Based on 698 testing images, we synthesize different levels of motion-blurred, Gaussian blurred, and out-of-focus images using the models introduced in Section 3.

The classification performance of these three blur-invariant approaches are shown in Fig. 14. We can see that the performance of them are quite stable when the blur level increases, which verifies the robustness of these approaches to image blurs. However, their performance (<30 percent accuracy) is much lower than the CNN-based approaches as reported in Fig. 12. This is mainly due to the usage of low-level image features in these blur-invariant approaches,

TABLE 5
The Rank Correlation Between Image-Classification Accuracy and PSNR/SSIM at Each Motion-Blur Level

Correlation	$l = 5$	$l = 10$	$l = 15$	$l = 20$	$l = 25$
(Accuracy, PSNR)	0.3	0.3	0.3	0.3	0.1
(Accuracy, SSIM)	0.3	0.1	0.3	0.3	0.3

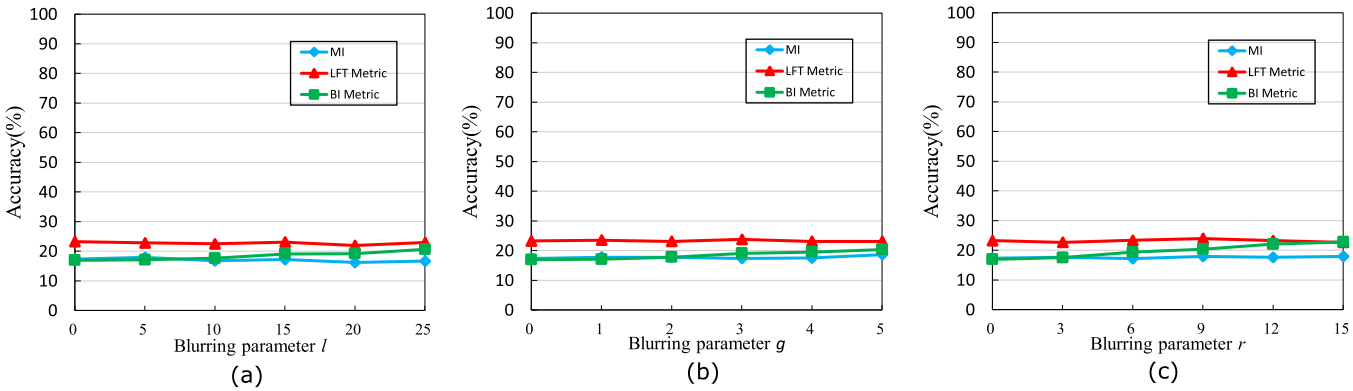


Fig. 14. Performance of blur-invariant approaches on a subset of Caltech-256 data, with different kinds and levels of blurs: (a) motion-blurred images, (b) Gaussian-blurred images, and (c) out-of-focus images.

which cannot handle the significant within-class variations, e.g., different object instances, varying view angles, different backgrounds etc, in image classification. For example, the class of 'dog' in Caltech-256 dataset contains images of different breeds of dogs with different appearances, poses, sizes, in different backgrounds. The CNN-based approaches can better handle their classification by supervisedly learning deep features from a variety of training data. The unsupervised low-level-feature-based blur-invariant approaches are more suitable for handling images collected in restricted environment where each object class shows small within-class variations, such as in the application of face recognition [81].

6 Conclusion

In this paper, we conducted an empirical study to explore the effect of nine kinds of image degradations to the performance of CNN-based image classification. For facilitating the quantitative evaluation, we synthesized a large number of images for training and testing, including hazy images, motion-blurred images, fish-eye images, underwater images, low resolution images, salt-and-peppered images, images with white Gaussian noise, Gaussian-blurred images and out-of-focus images, each with six degradation levels. We also collected a set of real hazy images from Internet. We found that the image classification performance does drop significantly when the image is degraded, especially when the training images can not well reflect the degradation levels of the test images. By visualizing the activations of hidden layers of the CNN classifiers, we found that many important low level features were not well discerned in early layers, which might be a key factor for the dropped classification accuracy. We also found that the existing algorithms for removing haze and motion-blur degradations could not improve the CNN-based classification performance much. We hope this study can draw more interests from the community to work on degraded image classification, which can benefit many important application domains such as autonomous driving, underwater robotics, video surveillance, and wearable cameras.

Acknowledgments

This work is supported, in part, by the Fundamental Research Funds for the Central Universities (2019JBZ104, 2016JBZ005), and National Natural Science Foundation of China (61906013, 51827813, 61672376, U1803264). The preliminary version of this work has been published in 2018 European Conference on Computer Vision [82].

References

[1] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1794–1801.
[2] J. Sánchez and F. Perronnin, "High-dimensional signature compression for large-scale image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1665–1672.
[3] 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.
[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 730–734.
[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 730–734.
[6] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.
[7] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. 7694, 2007.
[8] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.
[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[10] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
[11] T. Durand, T. Mordan, N. Thome, and M. Cord, "WILDCAT: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5757–5966.
[12] S. Karahan, M. K. Yildirum, K. Kirtac, F. S. Rende, G. Butun, and H. K. Ekenel, "How image degradations affect deep CNN-based face recognition?" in *Proc. Int. Conf. Biometrics Special Interest Group*, 2016, pp. 1–5.
[13] Z. Wang, S. Chang, Y. Yang, D. Liu, and T. S. Huang, "Studying very low resolution recognition using deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1–9.
[14] D. Liu, B. Cheng, Z. Wang, H. Zhang, and T. S. Huang, "Enhance visual recognition under adverse conditions via deep networks," *IEEE Trans. Image Process.*, vol. 28, no. 9, pp. 4401–4412, Sep. 2017.
[15] 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, Dec. 2011.
[16] J.-P. Tarel and N. Hautiere, "Fast visibility restoration from a single color or gray level image," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 2201–2208.
[17] J.-P. Tarel, N. Hautiere, A. Cord, D. Gruyer, and H. Halmaoui, "Improved visibility of road scene images under heterogeneous fog," in *Proc. IEEE Intell. Veh. Symp.*, 2010, pp. 478–485.

[18] 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.

[19] 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.

[20] D. Berman *et al.*, "Non-local image dehazing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1674–1682.

[21] 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.

[22] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang, "Single image dehazing via multi-scale convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016.

[23] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, "AOD-Net: All-in-one dehazing network," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4780–4788.

[24] W. Ren *et al.* "Gated fusion network for single image dehazing," in *Proc. Int. Conf. Comput. Anal. Images Patterns*, 2018, pp. 3253–3261.

[25] R. Li, J. Pan, Z. Li, and J. Tang, "Single image dehazing via conditional generative adversarial network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8202–8211.

[26] Y. Qu, Y. Chen, J. Huang, and Y. Xie, "Enhanced pix2pix dehazing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8160–8168.

[27] R. Liu, Z. Li, and J. Jia, "Image partial blur detection and classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[28] E. Kalalembang, K. Usman, and I. P. Gunawan, "DCT-based local motion blur detection," in *Proc. Int. Conf. Instrum, Commun., Inf. Technol., Biomed. Eng.*, 2009, pp. 1–6.

[29] J. Gast, A. Sellent, and S. Roth, "Parametric object motion from blur," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1846–1854.

[30] A. Rajagopalan and R. Chellappa, *Motion Deblurring: Algorithms and Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2014.

[31] J. Pan, D. Sun, H. Pfister, and M. H. Yang, "Blind image deblurring using dark channel prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1628–1636.

[32] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 257–265.

[33] S. Ramakrishnan, S. P. A. Gangopadhyay, and S. Raman, "Deep generative filter for motion deblurring," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, 2017, pp. 2993–3000.

[34] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "Deblurgan: Blind motion deblurring using conditional adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Patt. Recognit.*, 2018, 8183–8192.

[35] X. Tao, H. Gao, Y. Wang, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8174–8182.

[36] B. Lu, J.-C. Chen, and R. Chellappa, "Unsupervised domain-specific deblurring via disentangled representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10225–10234.

[37] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1335–1340, Aug. 2006.

[38] H. Fu and X. Cao, "Forgery authentication in extreme wide-angle lens using distortion cue and fake saliency map," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 4, pp. 1301–1314, Aug. 2012.

[39] C. Hughes, P. Denny, M. Glavin, and E. Jones, "Equidistant fish-eye calibration and rectification by vanishing point extraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2289–2296, Dec. 2010.

[40] O. Krams and N. Kiryati, "People detection in top-view fisheye imaging," in *Proc. IEEE Int. Conf. Adv. Video Signal Based Surveillance*, 2017, pp. 1–6.

[41] I. Baek, A. Davies, Y. Geng, Ragunathan, and Rajkumar, "Real-time detection, tracking, and classification of moving and stationary objects using multiple fisheye images," in *Proc. IEEE Intell. Vehicles Symp.*, 2018, pp. 447–452.

[42] K. Oliver, W. Hou, and S. Wang, "Feature matching in underwater environments using sparse linear combinations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2010, pp. 447–452.

[43] D. Akkaynak, T. Treibitz, T. Shlesinger, Y. Loya, R. Tamir, and D. Iluz, "What is the space of attenuation coefficients in underwater computer vision?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 568–577.

[44] N. Wang, B. Zheng, H. Zheng, and Z. Yu, "Feeble object detection of underwater images through LSR with delay loop," *Opt. Express*, vol. 25, no. 19, pp. 22 490–22 498, 2017.

[45] T. Möller, I. Nillsen, and T. W. Nattkemper, "Active learning for the classification of species in underwater images from a fixed observatory," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2891–2897.

[46] A. A. Rajeev, S. Hiranwal, and V. K. Sharma, "Improved segmentation technique for underwater images based on k-means and local adaptive thresholding," in *Proc. Inf. Commun. Technol. Sustainable Develop.*, 2018, pp. 443–450.

[47] Z. Chen, Z. Zhang, Y. Bu, F. Dai, T. Fan, and H. Wang, "Underwater object segmentation based on optical features," *Sens.*, vol. 18, no. 1, 2018, Art. no. 196.

[48] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[49] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2790–2798.

[50] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1664–1673.

[51] W. Han, S. Chang, D. Liu, M. Yu, M. Witbrock, and T. S. Huang, "Image super-resolution via dual-state recurrent networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1654–1663.

[52] X. Wang, K. Yu, C. Dong, and C. L. Chen, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 606–615.

[53] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2472–2481.

[54] Z. Hui, X. Wang, and X. Gao, "Fast and accurate single image super-resolution via information distillation network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 723–731.

[55] H. Dong, J. Yu, and C. Xiao, "Dual reweighted Lp-norm minimization for salt-and-pepper noise removal," 2018, *arXiv:1811.09173*.

[56] B. Fu, X. Zhao, Y. Li, and X. Wang, "Patch-based contour prior image denoising for salt and pepper noise," *Multimedia Tools Appl.*, vol. 77, no. 10, pp. 1–11, 2018.

[57] B. Fu, X. Zhao, Y. Li, X. Wang, and Y. Ren, "A convolutional neural networks denoising approach for salt and pepper noise," *Multimedia Tools Appl.*, vol. 77, no. 8, pp. 1–15, 2018.

[58] B. Fu, X. Zhao, C. Song, X. Li, and X. Wang, "A salt and pepper noise image denoising method based on the generative classification," *Multimedia Tools Appl.*, vol. 78, no. 9, pp. 12043–12053, 2019.

[59] Z. Kai, W. Zuo, S. Gu, and Z. Lei, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2808–2817.

[60] S. Lefkimmiatis, "Universal denoising networks: A novel CNN-based network architecture for image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3204–3213.

[61] J. Flusser, T. Suk, S. Farokhi, and C. Hschl, "Recognition of images degraded by Gaussian blur," in *Proc. Int. Conf. Comput. Anal. Images Patterns*, 2015, pp. 88–99.

[62] S. S. Al-amri *et al.*, "Deblured Gaussian blurred images," 2010, *arXiv:1004.4448*.

[63] N. Yair and T. Michaeli, "Multi-scale weighted nuclear norm image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2443–2452.

[64] K. Yu, C. Dong, L. Lin, and C. C. Loy, "Crafting a toolchain for image restoration by deep reinforcement learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018.

[65] W. Zhao, B. Zheng, Q. Lin, and H. Lu, "Enhancing diversity of defocus blur detectors via cross-ensemble network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8905–8913.

[66] C. Tang, X. Zhu, X. Liu, L. Wang, and A. Zomaya, "DeFusionNet: Defocus blur detection via recurrently fusing and refining multi-scale deep features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2700–2709.

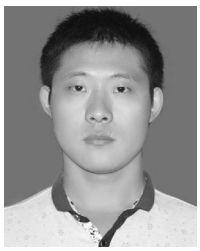
[67] J. Flusser, T. Suk, and S. Saic, "Recognition of blurred images by the method of moments," *IEEE Trans. Image Process.*, vol. 5, no. 3, pp. 533–538, Mar. 1996.

[68] J. Flusser and T. Suk, "Degraded image analysis: An invariant approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 6, pp. 590–603, Jun. 1998.
[69] Z. Zhang, E. Klassen, A. Srivastava, P. Turaga, and R. Chellappa, "Blurring-invariant riemannian metrics for comparing signals and images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1770–1775.
[70] R. Gopalan, S. Taheri, P. Turaga, and R. Chellappa, "A blur-robust descriptor with applications to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1220–1226, Jun. 2012.
[71] Z. Zhang, E. Klassen, and A. Srivastava, "Gaussian blurring-invariant comparison of signals and images," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3145–3157, Aug. 2013.
[72] J. Zhang, Y. Cao, S. Fang, Y. Kang, and C. W. Chen, "Fast haze removal for nighttime image using maximum reflectance prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7016–7024.
[73] S. Jian, W. Cao, Z. Xu, and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 769–777.
[74] L. Dolin, G. Gilbert, I. Levin, and A. Luchinin, "Theory of imaging through wavy sea surface," Russian Acad. Sci., Inst. Appl. Phys., Nizhniy Novgorod, Russia, 2006.
[75] D. Chao, C. L. Chen, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 184–199.
[76] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 105–114.
[77] M. E. Moghaddam, "A mathematical model to estimate out of focus blur," in *Proc. Int. Symp. Image Signal Process. Anal.*, 2007, pp. 278–281.
[78] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electron. Lett.*, vol. 44, no. 13, pp. 800–801, 2008.
[79] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[80] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5188–5196.
[81] L. Kuang-Chih, H. Jeffrey, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 684–698, May 2005.
[82] Y. Pei, Y. Huang, Q. Zou, Y. Lu, and S. Wang, "Does haze removal help CNN-based image classification?" in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 682–697.

Yanting Pei is currently working toward the PhD degree in the School of Computer and Information Technology, Beijing Jiaotong University, China. In 2018, she was a visiting PhD Student with the University of South Carolina, Columbia, South Carolina. Her current research interests include computer vision, image processing, machine learning, and deep learning.

Yaping Huang received the BS, MS, and PhD degrees from the School of Computer and Information Technology, Beijing Jiaotong University, China, in 1995, 1998, and 2004, respectively. She is currently a professor with the School of Computer and Information Technology, Beijing Jiaotong University. Her research interests include computer vision, machine learning, and pattern recognition.

Qi Zou received the BS and PhD degrees from Beijing Jiaotong University, China, in 2001 and 2006, respectively. She is currently a professor with the School of Computer and Information Technology, Beijing Jiaotong University. Her research interests include pattern recognition, computer vision, computer vision, and target tracking.



Xingyuan Zhang is working toward the PhD degree in the School of Computer and Information Technology, Beijing Jiaotong University, China. In 2019, he was a visiting PhD student with the University of South Carolina, Columbia, South Carolina. His research interests include deep learning, computer vision, and digital image processing.

Song Wang received the PhD degree in electrical and computer engineering from the University of Illinois at UrbanaChampaign (UIUC), Champaign, Illinois, in 2002. He was a research assistant with the Image Formation and Processing Group, Beckman Institute, UIUC, from 1998 to 2002. In 2002, he joined the Department of Computer Science and Engineering, University of South Carolina, Columbia, South Carolina, where he is currently a professor. His current research interests include computer vision, image processing, and machine learning. He is currently serving as the publicity/web portal chair of the Technical Committee of Pattern Analysis and Machine Intelligence of the IEEE Computer Society, an associate editor of the *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *Pattern Recognition Letters*, and *Electronics Letters*. He is a senior member of the IEEE and a member of the IEEE Computer Society.