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

A ssociating 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 quantitiy 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...
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

![Degraded Images Examples](image)

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.
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 L1 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-pepper 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 blur. 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. Specifically, 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 $\beta$. 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 $c$. 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 $r$ and the single scattering albedo $\omega$. We follow [75], [76] to construct low-resolution images by down-sampling the high-resolution images using the bicubic kernel with down-sampling 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
accuracy $\text{Accuracy} = \frac{R}{N}$ 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.

### 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...
For underwater images, we set parameters \( \tau \in \{0, 0.6, 0.7, 0.8, 0.9, 1, 1.1\} \). For low resolution images, we set the parameter \( f \in \{2, 4, 6, 8, 10\} \). 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.

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.
achieved by training and testing on the image data 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 (β = 0) to train CNN classifiers, and then apply them to classify images with hazy level of β = 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 β = 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 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.

![Image](image.png)

**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
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-blurred 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, 79.5, 78.4, 76.7, 73.9, 69.5, 77.9, 76.7, 74.5, 71.8, 68.9, 66.4, 78.7, 77.6, 76.5, 74.7, 73.1, 70.4, 80.2, 79.9, 79.7, 79.2, 77.9, 76.0, 76.2, 73.9, 68.4, 62.4, 56.4, 51.0, 76.9, 75.5, 73.9, 72.7, 71.2, 69.5, 74.8, 71.4, 69.3, 67.4, 66.1, 64.3, 78.8, 78.0, 75.5, 73.0, 70.0, 67.0, 77.0, 74.6, 70.8, 67.0, 63.5, 59.4, respectively. This trend is largely consistent with what we find from the above synthetic image experiments.

### 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 pool1, pool2, pool3, pool4, and pool5, 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.

From Table 2, we can see that the classification accuracies of clear images in its nine subtables are 79.7, 79.9, 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 2

<table>
<thead>
<tr>
<th>hazy images: $\beta$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>79.7</td>
<td>79.5</td>
<td>78.4</td>
<td>76.7</td>
<td>73.9</td>
<td>69.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>motion-blurred images: $l$</th>
<th>0</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>77.9</td>
<td>76.7</td>
<td>74.5</td>
<td>71.8</td>
<td>68.9</td>
<td>66.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>fish-eye images: $e$</th>
<th>$\frac{1}{4}$</th>
<th>1.25</th>
<th>1.5</th>
<th>1.75</th>
<th>2</th>
<th>2.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>78.7</td>
<td>77.6</td>
<td>76.5</td>
<td>74.7</td>
<td>73.1</td>
<td>70.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>underwater images: $r$, $o$</th>
<th>(0,0)</th>
<th>(8,0.6)</th>
<th>(9,0.7)</th>
<th>(10,0.8)</th>
<th>(11,0.9)</th>
<th>(12,1,0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>80.2</td>
<td>79.9</td>
<td>79.7</td>
<td>79.2</td>
<td>77.9</td>
<td>76.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>low resolution images: $f$</th>
<th>0</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>76.2</td>
<td>73.9</td>
<td>68.4</td>
<td>62.4</td>
<td>56.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>salt-and-peppered images: $s$</th>
<th>0</th>
<th>0.05</th>
<th>0.1</th>
<th>0.15</th>
<th>0.2</th>
<th>0.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>76.9</td>
<td>75.5</td>
<td>73.9</td>
<td>72.7</td>
<td>71.2</td>
<td>69.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>images with white Gaussian noise: $v$</th>
<th>0</th>
<th>0.05</th>
<th>0.1</th>
<th>0.15</th>
<th>0.2</th>
<th>0.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>74.8</td>
<td>71.4</td>
<td>69.3</td>
<td>67.4</td>
<td>66.1</td>
<td>64.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gaussian-blurred images: $g$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>78.8</td>
<td>78.0</td>
<td>75.5</td>
<td>73.0</td>
<td>70.0</td>
<td>67.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>out-of-focus images: $r$</th>
<th>0</th>
<th>3</th>
<th>6</th>
<th>9</th>
<th>12</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>77.0</td>
<td>74.6</td>
<td>70.8</td>
<td>67.0</td>
<td>63.5</td>
<td>59.4</td>
</tr>
</tbody>
</table>

These results indicate the importance of considering a wide range of image degradation levels in training CNNs. Additionally, the performance drop observed when training with real hazy images highlights the need for robust training strategies that can handle such variations. The hidden-layer features analysis provides insights into why these performance drops occur, offering potential avenues for improving degradation robustness in CNNs.
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 haze removal and motion-blur removal 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 curves 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.
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% / 69.5%, 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
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.

5.2 Does Motion-Blur Removal Help 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 $l = 5, 10, 15, 20, 25, 25$, we rank all the deblurring methods based on the classification accuracy. We then rank these methods based on average PSNR and

<table>
<thead>
<tr>
<th>Correlation</th>
<th>$\beta = 1$</th>
<th>$\beta = 2$</th>
<th>$\beta = 3$</th>
<th>$\beta = 4$</th>
<th>$\beta = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Accuracy, PSNR)</td>
<td>0.3455</td>
<td>-0.0545</td>
<td>0.4061</td>
<td>0.0667</td>
<td>0.2485</td>
</tr>
<tr>
<td>(Accuracy, SSIM)</td>
<td>-0.0909</td>
<td>-0.2970</td>
<td>-0.3333</td>
<td>0.1636</td>
<td>-0.2606</td>
</tr>
</tbody>
</table>

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.
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. 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

<table>
<thead>
<tr>
<th>Correlation</th>
<th>$l = 5$</th>
<th>$l = 10$</th>
<th>$l = 15$</th>
<th>$l = 20$</th>
<th>$l = 25$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Accuracy, PSNR)</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>(Accuracy, SSIM)</td>
<td>0.3</td>
<td>0.1</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
</tbody>
</table>
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

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 Urbana-Champaign (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 Transactions 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.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.