Two-Class Weather Classification

Cewu Lu, *Member, IEEE*, Di Lin, Jiaya Jia, *Senior Member, IEEE*, and Chi-Keung Tang, *Senior Member, IEEE*

Abstract—Given a single outdoor image, we propose a collaborative learning approach using novel weather features to label the image as either sunny or cloudy. Though limited, this two-class classification problem is by no means trivial given the great variety of outdoor images captured by different cameras where the images may have been edited after capture. Our overall weather feature combines the data-driven convolutional neural network (CNN) feature and well-chosen weather-specific features. They work collaboratively within a unified optimization framework that is aware of the presence (or absence) of a given weather cue during learning and classification. In this paper we propose a new data augmentation scheme to substantially enrich the training data, which is used to train a latent SVM framework to make our solution insensitive to global intensity transfer. Extensive experiments are performed to verify our method. Compared with our previous work and the sole use of a CNN classifier, this paper improves the accuracy up to 7-8 percent. Our weather image dataset is available together with the executable of our classifier.

Index Terms—Weather understanding, image classification, structure SVM

1 Introduction

We address the problem of two-class weather classification from a single outdoor image. This seemingly easy task for humans—to tell whether a given image is sunny or cloudy—turns out to be challenging. This paper attempts to provide technical insight and solutions to address the above issues, while acknowledging that our work is a first but significant step for weather understanding from single images. Note that naive schemes based on image brightness or color/intensity statistics (Figs. 1 and 2) are doomed to fail in this two-class classification problem. While hardware solutions relying on expensive sensors are employed, for centuries human vision is still the most powerful tool for weather observation. If we can exploit existing surveillance and smartphone cameras, which are found almost everywhere, it may be possible to turn human weather observation into a powerful and cost-effective computer vision application.

Previously, in [37], we demonstrated that careful engineering of well-chosen weather-specific features employed in supervised learning can adequately address this two-class weather classification problem. In this paper, we further investigate the efficacy of the state-of-the-art convolutional neural network (CNN) in solving the problem. The CNN approach as well as the CNN feature is data-driven, which is in contrast to hand-picked weather features in [37]. As will be shown in the experimental section, we found that the concatenation of the CNN feature and weather-specific features reports the best performance, namely, a 7-8 percent improvement over the sole use of a CNN classifier that is trained end-to-end using the given training data. The data-driven CNN feature and the weather-specific features work together to exploit the synergies between the two. Based on this new overall weather feature, our approach consists of the following three technical contributions:

First, we describe the design and implementation of various weather cues, which are used to form the weather feature [37]. These everyday weather cues (such as sky, shadow, reflection, contrast and haze) are what humans are still using for weather observing—a hazy or grayish sky characterizes a cloudy day while hard shadow cast on the ground indicates a sunny day, as illustrated in Fig. 3a Conversely, in the absence of any weather cues, even we humans may not be confident in labeling the weather type, as illustrated in Fig. 3b. In this paper, we concatenate the CNN feature with the above weather-specific features to form the overall weather feature in training and testing.

Given the overall weather feature, the next question is how to properly learn the classifier. The main issue is that the weather cues used in this paper may not be all available in an image—e.g., not every outdoor image has a sky region—which is problematic to a discriminative training process adopted by traditional classifiers such as SVM. To address this issue, our second technical contribution consists of a *collaborative learning* framework using homogeneous voters—the outdoor images are clustered where images in the same cluster are similar in terms of weather cues. This allows us to build a classifier in a conventional way thanks to the homogeneity in each cluster. The final labeling is the weighted voting result of the cluster classifier outputs. The cluster closer to the

- *C. Lu is with the Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai 200240, China. E-mail: afflucewu06@gmail.com.*
- *D. Lin is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong 518060, China. E-mail: ande.lin1988@gmail.com.*
- *J. Jia is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong. E-mail: leojia@cse.cuhk.edu.hk.*
- *C.-K. Tang is with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology. E-mail: cktang@cse.ust.hk.*

Manuscript received 4 May 2016; revised 4 Sept. 2016; accepted 22 Oct. 2016. Date of publication 0 . 0000; date of current version 0 . 0000.
Recommended for acceptance by D. Xu.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2016.2640295

(a) (b)

Fig. 1. (a) A sunny image with mean lightness 32.41. (b) A cloudy image with mean lightness 58.25.

testing image is given a higher weight. As will be explained in the following, homogeneous voters are learned under a unified optimization framework.

To make our system more robust to training images harvested from the web, we propose a novel strategy to enrich our training set by synthesizing for each training image its *weather counterpart images*, which belong to a subclass of images of the same scene taken under different camera settings and/or after photo editing that can be characterized by *a global color/intensity transfer*. The training image and its synthesized weather counterparts together are then used in *latent SVM learning* which encourages each training sample and its counterparts to have the *same* weather label. Our synthesis strategy is scalable. That is, the production of the weather counterparts given a training image is fully automatic, which requires no further data collection or annotation by humans.

Finally, we perform quantitative comparison with a number of typical baselines including SVM, Adaboost [58], [62], and prior weather-related methods [26], [46], [61]. Our final contribution consists of a 10,000-image weather dataset in which the images are properly selected and annotated. This is used to evaluate our learning and labeling strategy.

This manuscript extends its conference version [37] along the following dimensions:

- The overall weather feature combines the data-driven CNN feature and handcrafted weather features.
- A data-driven approach for synthesizing weather counterparts to make scalable data collection and training; the new system is insensitive to global intensity transfer and achieves improvement over [37].
- A latent SVM framework is proposed to capture a wide variety of global intensity transfer.
- More experiments are conducted to evaluate the proposed method.

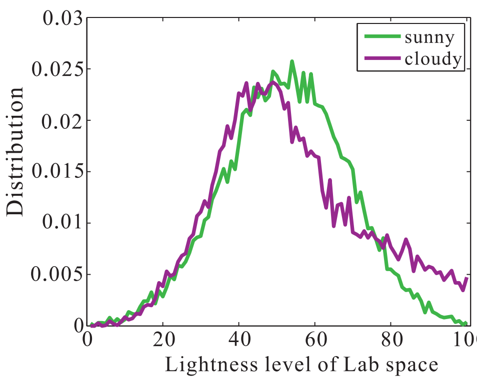
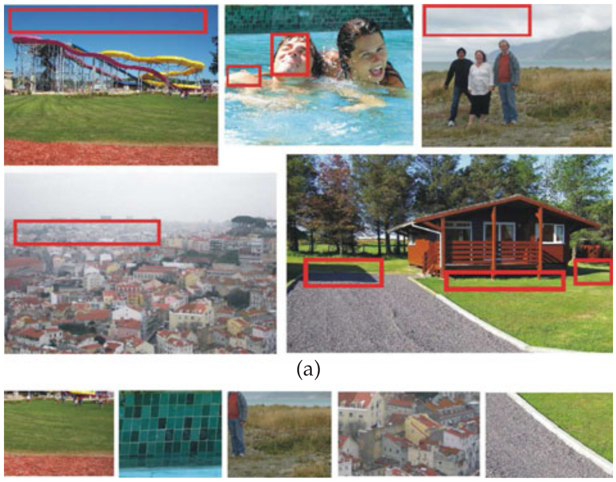


Fig. 2. Pixel intensity distributions in the lightness L channel in the LAB color space of 5 K cloudy images and 5 K sunny images. It is almost impossible to draw a decision boundary between the two types of weather.



(a)

(b)

Fig. 3. Weather cues. (a) Common weather cues in red rectangles. (b) Regions in (a) lacking any weather cues.

The paper is organized as follows. In Section 2 we review the related work. In Section 3, we introduce our weather-specific and data driven CNN features. In Section 4, we describe our weather dataset and weather counterparts generation. Section 5 presents the collaborative and latent SVM learning of our weather classifier. Section 6 discusses our results on weather classification. We conclude this paper in Section 7.

2 Related Work

This section gives an overview of the related work on weather understanding, which can be regarded as a category in scene recognition. The background of the convolutional neural network is also investigated since the CNN feature is an important component in our computational framework.

2.1 Weather Understanding

2.1.1 *Weather Understanding with Hand-Crafted Feature*

Weather understanding plays a vital role in many real-world applications such as navigation control in self-driving cars. Automatic understanding of weather conditions enhances road safety by, for instance, controlling the vehicle speed in response to real-time weather situation [46], [61]. In [20], a built-in weather understanding component was found in an accurate navigation system that involves sky detection. Raindrops have been a frequently used cue for weather recognition, and in [23], discriminative raindrop templates were learned to infer weather situation. In [50], a photometric stereo-based method was proposed to estimate weather situation. Multiple images were required to estimate the illumination situation of a given site. Therefore, only a few sites (e.g., popular tourist sites) can meet this requirement. In [40] Narasimhan et al. proposed a physics-based model to capture multiple scattering of light rays from a source to the camera. This model works well for cases where scattering effect is strong, such as fog, haze, mist and rain at night. In [24], 40 transient attributes were studied and a model was learned to predict these attributes given a single image. The approach is standard "features + SVM" scheme where the features used are standard (e.g., HOG, SIFT) which may not capture well weather characteristics. In [65] a multi-class weather classification method was described using multiple weather features and multiple kernel learning.

In [7], [51], weather types were recognized by the motion of cloud, snow flakes, etc. These weather-related phenomena depict periodic movement which provides specific weather patterns for detection.

Though the above methods have shown good performance in their respective applications, custom devices or conditions were often required. Exploiting existing smartphones which are cheaper, and surveillance cameras which can be found or installed almost everywhere, can make it possible to turn general weather observation into a powerful and cost-effective computer vision application.

2.1.2 Weather Understanding with CNN

Our problem is related to classification which can also be solved by Convolutional Neural Networks. Deep CNN has led a series of breakthroughs in image classification [22]. It is a feed-forward, end-to-end multilayered neural network inspired by the organization of the animal visual cortex. The convolutional neural network has found applications in image and video recognition tasks, such as video classification [19], object detection [11], and action recognition [1]. Excellent models, such as AlexNet [22], Network-in-Network [35], VGG [52] and ResNet [14] have been developed. Unlike general image classification (e.g., object classification), weather classification relies on weather-sensitive cues (as we shall demonstrate in the experiment section) which is somewhat similar to fine-grained recognition in the level of details required, while the deep CNN is excellent in capturing global scene semantics which needs to be integrated to make such recognition succeed [34], [64]. There is only a handful of research attempts in applying CNN to weather understanding. In [9], convolutional neural networks (VGG) is directly used in classifying weather, and in [56] to predict outdoor ambient temperature and the time of the year. The CNN filter may however miss subtle weather cues inherent in the input image.

2.2 Scene Understanding

Weather understanding is a specific case of scene recognition. General scene recognition focuses on discovering discriminative scene structure. As explained in [45], scene structure can be regarded as a combination of parts which are called regions of interest. These discriminative parts provide a powerful representation of the scene. Thus exploiting them in relevant tasks has recently become a popular trend.

The work [17], [30], [31], [32], [48], [53] discovered parts with specific visual concepts. Each learned part is expected to represent a single or a cluster of visual objects, which is beneficial to alleviate visual ambiguity. Meanwhile, unsupervised discovery of discriminative parts has received much attention. Though handcrafted part filters are easy to comprehend, they strongly rely on human labeling and are not scalable. Unsupervised frameworks [18], [21], [28], [32], [33], [42], [43], [49], [55], [66] can be more practical and efficient especially for large data sets.

Recently, various mid-level representations have been employed to enhance the discriminative power in classification [3], [30], [31], [66]. State-of-the-art methods [17], [32], [53] proposed discriminative parts and used them to construct mid-level representation, e.g., response maps obtained from convolution with part filters. These mid-level representations are fed into discriminative classifiers and evaluated on different scene classification datasets. Mid-level representation can be a better alternative or complementary to traditional low-level representations [6], [27], [36], [41], [44], [58], [62], because mid-level representation is capable of differentiating among a large variety of inter and intra categories as described in [63].

We note the methods in scene classification cannot be employed to solve our problem. Though weather is part of the scene, it is not as concrete (e.g., no closed boundary) as objects such as trees, buildings, and mountains, thus risking information loss when we apply these methods.

2.3 Weather Applications

Weather cues have been used to enable various applications. In [2], deep convolutional neural networks was used to estimate transient attributes including weather, time of the day, season and subjective properties of a given scene. In [12], the interaction between the appearance of an outdoor scene and the ambient temperature was studied, where the statistical correlations between image sequences from outdoor cameras and temperature measurements were derived. In [15], weather conditions with the scene structure and position of the sun were used to estimate the time and location the image was captured. Snow recognition was studied in [57], which enables the production of satellite maps of snowfall using geo-tagged, time stamped images from Flickr. Cloud cues were explored in [59], where a method was presented for estimating the geometry of an outdoor scene. Another work that used cloud cues is [16], where cloud motion enables geometric calibration of static outdoor cameras.

3 The Overall Weather Feature

We compute for each image the *overall weather feature*, a 4717-D feature consisting of two parts, namely the CNN feature and five weather features. The feature vector is formed by concatenating the six components

$$[\mathbf{f}_{sk};\ \ \mathbf{f}_{sh};\ \ \mathbf{f}_{re};\ \ \mathbf{f}_{co};\ \ \mathbf{f}_{ha};\ \ \mathbf{f}_{cn}], \tag{1}$$

where the first five features, namely, sky, shadow, reflection, contrast and haze, correspond to a key weather cue to be defined shortly. We incorporate the CNN feature [22] $\mathbf{f}_{cn}$ to describe the image in general, which is extracted from a learned two-class weather CNN model. Since not all of these cues are necessarily present in a given outdoor image, we also compute the *existence vector*

$$[v_{sk};\ \ v_{sh};\ \ v_{re};\ \ v_{ha};\ \ v_{cn}], \tag{2}$$

where each scalar score in $[0, 1]$ indicates the confidence that the corresponding weather cue is present in the given image and in particular, v_{cn} is the confidence score of the CNN classifier. Since image contrast difference exists in both sunny and cloudy photos, v_{co} is always 1 and excluded.

3.1 Weather Feature

3.1.1 Sky

If present, the sky is the most important cue for weather labeling. A clear, cloudless sky is blue as air molecules

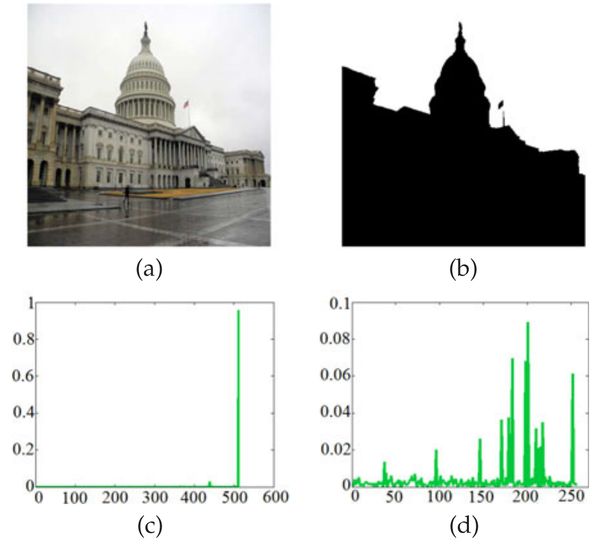


Fig. 4. Sky. (a) input image, (b) detected sky region, (c) color histogram of the sky, (d) plot of $\mathbf{f}_{sk}$.

scatter blue light more than red light. Cloud is made of tiny water droplets which make the sky look grayish white.

To define v_{sk}, the sky region is detected in a pixel-wise manner in the following steps. We respectively collect 20,000 sky and non-sky patches, each of size 15×15, and extract a 131 dimensional feature, which contains the SIFT descriptor (128D) and mean HSV color (3D). This feature was suggested in [54]. Then a random forest classifier is learned on the two patch classes. Now, given an image, we uniformly sample 15×15 patches and test their labels (sky or non-sky) as seeds. Sky region can be segmented by implementing graph cuts on those seeds (see Figs. 4a and 4b). Let A be the sky to image area ratio. We set $v_{sk} \in [0, 1]$ as

$$v_{sk} = \begin{cases} 1 & \text{if } A > 0.5 \\ \min\{2A, 1\} & \text{otherwise.} \end{cases} \quad (3)$$

To define the $\mathbf{f}_{sk}$ vector we have considered various alternatives. Straightforward color histogram feature in the sky region suffers from two defects. First, possible sky colors (both cloudy and sunny) are sparse, thus yielding most color bins with the zero value (Fig. 4c). Second, no adequate consideration is given to color contrast. In this paper, we define $\mathbf{f}_{sk}$ using color-pair dictionary coding as follows.

We collect 2,000 images with detected sky regions. Neighborhood pixels in pairs are extracted from the sky region to form a large number of 6D vectors, each of them consisting of a total of 6 RGB values. This process results in about 100,000 pixel pairs. We then learn a sky color-pair dictionary $\mathbf{D} \in \mathbb{R}^{6\times 256}$ on the vectors using the method described in [38], thus producing a set of neighborhood-pixel vectors sparsely coded over the learned dictionary, expressed as

$$\min_{\beta_i} \|p_i - \mathbf{D}\beta_i\|_2^2 + \lambda\|\beta_i\|_1, \quad (4)$$

where $p_i \in \mathbb{R}^{6\times 1}$ is the ith vector, $\beta_i \in \mathbb{R}^{256\times 1}$ is the sparse code over $\mathbf{D}$. We solve Eq. (4) using [58]. Our final $\mathbf{f}_{sk}$ is filtered by max pooling of all β_i. That is, the jth bin of our feature is set to $\max_i\{\beta_{i,j}\}$ where $\beta_{i,j}$ is the jth bin of β_i.

Max pooling can preserve subtle sun-to-cloud contrast in the feature representation. Fig. 4d shows a typical $\mathbf{f}_{sk}$ plot.

Fig. 5. Shadow detection results of [25] for (a) a cloudy image and (b) a sunny image. Shadow detection in cloudy images is vulnerable to false detection.

In comparison to color histogram, our 256-D $\mathbf{f}_{sk}$ covers the full range of the histogram and encodes color contrast information as well. The advantage over color histogram was demonstrated in [37].

3.1.2 Shadow

Hard shadow boundaries form another useful cue because they are often found in outdoor photos shot in sunny days. To compute v_{sh} and $\mathbf{f}_{sh}$, we resort to shadow detection tools. Unlike sky detection, shadow detection in an image is still a challenging problem. Our extensive evaluation indicates while working well in sunny images, state-of-the-art shadow detection often fails for cloudy images, where dark regions are often misclassified as shadow as shown in Fig. 5.

Notwithstanding, we apply [25], rank the resulting shadow boundary confidence scores and take the 10th highest score to set v_{sh}. This serves as a rough relative indicator in our method. A larger v_{sh} represents possibly stronger shadow presence. High precision is not needed in the estimation.

Using a data-driven approach we design our $\mathbf{f}_{sh}$ by relying on the shadows detected in the training images restricted to sunny outdoor photos. If a given boundary is similar to those training shadow boundaries, we regard this as a shadow boundary typical of a sunny image.

In detail, initially, for all of the sunny images in the training set, we apply [25] to detect shadow boundaries and generate their corresponding confidence scores and boundary descriptors. For each image, we keep only the top 10 most confident shadow boundaries, and save them to the pool $\mathcal{P}$ which has $10\,V$ samples, where V is the number of sunny images in the training set.

Given a boundary, we measure its likelihood to be a shadow boundary typical of a sunny photo by the mean distance to its K-nearest ($K = 5$) neighbors in $\mathcal{P}$. Two examples of K-nearest neighbor matching are shown in Fig. 6. The Euclidean distance between the two boundaries descriptor vectors was used [25]. Given an image, we obtain its top 10

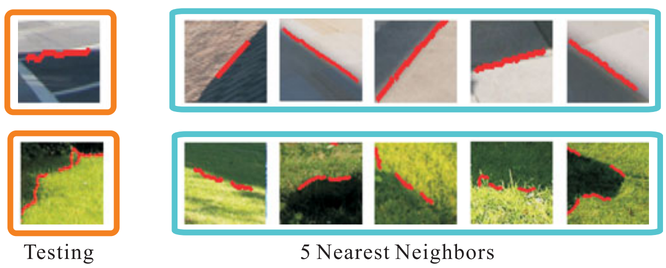


Fig. 6. K-nearest neighbor matching in $\mathcal{P}$. Shown in the blue rectangles are the five nearest neighbors.

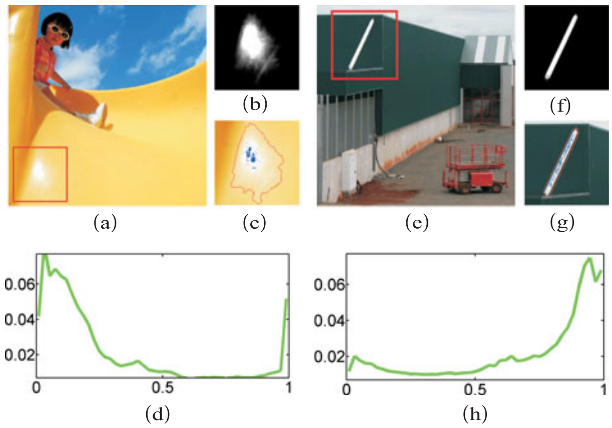


Fig. 7. Reflection cue. A sunny image with strong sunlight reflection in (a) versus a cloudy image with inherently white regions in (e). (b) and (f) are the corresponding alpha mattes. In (c) and (g), red and blue points indicate background and foreground seeds used in alpha matting. (d) and (h) are distributions of the alpha maps, taken as the $\mathbf{f}_{re}$ cue.

most confident shadow boundaries and compute their likelihood as described above to form the 10-D $\mathbf{f}_{sh}$ vector.

3.1.3 Reflection

Strong sunlight reflected from shiny objects is another powerful cue. Except for a perfect mirror reflector, sunlight reflection is usually characterized by a brightly lit region in the image where pixels in the region center are brightest and saturated in nearly all color channels. The reflection intensity decays from the center toward the boundary of the reflection region. An example was shown in Fig. 7, which compares strong sunlight reflection with the reflection from a white matte/dull object.

We set v_{re} to 1 if white pixels are present in the image and 0 otherwise. To construct $\mathbf{f}_{re}$, we apply image matting [29] at the detected white pixels. The definite foreground region consists of white pixels, and definite background region consists of a closed curve enclosing the foreground seeds. We then estimate the closed curve under the constraint that the euclidean distance between pixels along the curve and enclosed foreground seeds should be larger than a threshold (0.5 in our experiments). This closed curve can be computed by simple dynamic programming. An example was shown in Figs. 7b and 7c.

Given the matting result (e.g., Figs. 7b and 7f) we plot the alpha matte distributions as shown in Figs. 7d and 7h, and then assign the 100-bin alpha matte histogram as our 100-D $\mathbf{f}_{re}$ vector.

3.1.4 Contrast

Outdoor images captured in sunny and cloudy days exhibit different global and local saturation contrast. To compute $\mathbf{f}_{co}$, we utilize contrast information encoded as the percentile in image saturation. For example, a value at the 20th saturation percentile means that 20 percent of the image pixels are grayer. Clearly, if all saturation percentiles are the same for a given image, the saturation contrast is low. If on the other hand the 50th percentile is at 100 (saturation level) while the 49th percentile is 0, this image is very likely to have a high saturation contrast. In our paper, we use the C channel of LCH color space as our saturation map.

We collect all saturation percentile ratios to build $\mathbf{f}_{co}$ and leave the selection process to the final classifier. Specifically, we denote p_i as the ith percentile in the saturation map. The set of all saturation percentile ratios is given by $\{r|r = p_i/p_j, \forall i > j\}$, where i and j are multiples of 5. We thus obtain 171 percentile ratios in total, which are used to form our 171-D $\mathbf{f}_{co}$ vector. An example is shown in [37].

3.1.5 Haze

Cloudy weather may come with haze. Haze priors have been well studied in computer vision: the dark channel prior presented in [13] is effective. Similarly, we compute the dark channel as

$$\mathcal{J}^k(x) = \min_{r,g,b}\{\min_{y\in\Omega(x)}\{\mathcal{J}^c(y)\}\}, \tag{5}$$

where $\mathcal{J}^c$ is a color channel and $\Omega(x)$ is a local patch (with 8×8) centered at x. Most haze-free regions have a low intensity in the dark channel. We measure the haze level and set v_{ha} of a given image as the median value of its dark channel.

We define the $\mathbf{f}_{ha}$ component with the consideration that haze becomes thicker when a region is distant from the camera. These regions commonly exist at the top of an outdoor image. We consider haze location by using a spatial pyramid scheme. The input image is resized into 512×512. The dark channel in each image is uniformly partitioned into 2^2, 4^2, and 8^2 non-overlapping regions to obtain 84 sub-regions. We use the median value of the dark channel intensity in these regions to form the 84-D $\mathbf{f}_{ha}$ vector. An example of haze feature is shown in [37].

3.2 CNN Feature

Our new method in this paper includes the CNN feature which incorporates global discriminative information of the image. We train a CNN model on the two-class image set; the model we used is the AlexNet [22]. As in standard setting [11], the 7th layer neurons of the AlexNet model is extracted to form our 4096D feature $\mathbf{f}_{cn}$. This feature is effective when the CNN classifier is confident with its prediction. With the confidence scores of sunny and cloudy image s_{cnn} and c_{cnn} ($s_{cnn} + c_{cnn} = 1$), we measure v_{cn} as

$$v_{cn} = 2\max\{s_{cnn}, c_{cnn}\} - 1, \tag{6}$$

where v_{cn} is in the range of $[0, 1]$. Thus, only discriminable feature of the CNN model contributes to the system.

4 Weather Images and Counterparts Dataset

We created a new weather dataset that contains 10,000 images for training and testing. The dataset and the classifier executable are publicly available. The training images collected from the web were taken by different cameras, under different settings, and might have been edited as well. As most of our proposed weather features are designed based on scene illumination, we describe a new strategy to make our system insensitive to camera settings and photo editing that can be defined by a global transfer function.

(a) ISO = 200 (b) ISO = 800 (c) ISO = 1600 (d) (e) (f)

Fig. 8. Camera parameters and photo editing. (b) and (c) are weather counterparts of (a). These weather counterparts are obtained by adjusting camera parameters while capturing the same scene. Here, a high ISO value makes the picture sensitive to light. (e) and (f) are weather counterparts of (d). These weather counterparts are obtained via editing the original photos.

A robust weather recognition system should output a uniform weather label for an image as well as its *weather counterpart images*, which are images of the identical scene transformed by a global intensity mapping, and hence they should have the same weather label. Fig. 8 shows examples of weather counterpart images.

To make our data collection scalable, we propose to automatically generate weather counterparts for the training data. In the following, we first describe the construction of the weather dataset, and then present a perceptual study to validate the dataset, followed by detailing our learning-based weather counterpart generation.

4.1 Weather Image Dataset

Our weather dataset contains sunny and cloudy images obtained from three sources: Sun Dataset [60], Labelme Dataset [47] and Flickr. The minimum and maximum dimensions of the images are respectively 600 and 1,500.

To avoid bias, the helpers recruited to collect and label images were unaware of the purposes or methods used in our experiments. They worked with their own understanding, and collected each 14,000 outdoor images, in which sunny and cloudy images are in equal proportion.

We discarded very similar images by first computing the color histogram distance for all of the image pairs, and then rejected those identical or highly similar. As a result, 1,121 sunny images and 812 cloudy images were rejected. Next, we asked two helpers to independently check the remaining images (5,879 sunny and 6,188 cloudy). Images labeled as ambiguous weather condition by either or both of the helpers were discarded. A total of 5,467 sunny images and 5,612 cloudy images remained after this round. Finally, we asked the third helper to pick 5,000 sunny and 5,000 cloudy images in the final dataset.

4.2 Perceptual Validation

We first validate our weather image dataset using a user study. A total of 11 participants were recruited for the validation experiment. The participants consisted of five males and six females whose ages ranged from 26 to 41. All of the subjects reported normal or corrected-to-normal vision with no color-blindness, and reported that they were familiar with the outdoor scenes to be tested in the study. The participants were volunteers who were unaware of the purpose of the experiment.

We asked the participants to assign a sunny/cloudy score to each image where the two assigned scores should sum up to 1. We found that 97.6 percent of the images were assigned a score of 1 for the designated class by all of the subjects, which indicates that the weather type of most of the images in our dataset are unambiguous.

For the remaining 3.4 percent images, we compute the mean and variance of the user-assigned "probability scores" across different subjects, and report the average scores in Table 1. If we take 0.5 as the threshold for the "mean probability," the user-assigned weather type has 100 percent accuracy. We also regress these user-assigned probability scores, and the regression errors are shown in Table 2. Note on the one hand in the evaluation in the following sections, we do not use these user-assigned probabilities as the measurement metric due to the fact that the assigned scores are subjective, although the average score across different subjects is used here. On the other hand, we believe this may lead to interesting and worthwhile future work on using user-assigned probability scores (i.e., user's observation) in performance evaluation.

4.3 Weather Counterpart Image Generation

Now, we build a dataset for learning weather counterpart images. We capture 1,000 outdoor images with different camera parameters. To guarantee pixel-wise alignment, a tripod was used during capture. For each image, four weather counterpart images were captured using different camera parameters. The ISO setting is the most important camera parameter and we found it sufficient to capture a large set of tone variances in different cameras.

In addition, we recruited seven helpers to edit the 1,000 outdoor images using Photoshop. Five of them were without

TABLE 1
The Mean and Variance of the Mean User-Assigned Probabilities Not Equal to 0 or 1

	mean (Average)	variance (Average)
Sunny	0.88	0.03
Cloudy	0.85	0.02

TABLE 2
Regression Error (Mean $\pm$ Variance) of Different Methods

	Average regression error
LLC [58]	0.442 ± 0.027
ScSPM [62]	0.437 ± 0.031
CNN classifier [22]	0.032 ± 0.002
Ours	$\mathbf{0.026 \pm 0.003}$

The momentum, weight decay and learning rate of CNN are 0.0001, 0.9 and 0.005 respectively. The SVM step in LLC and ScSPM is replaced by SVR.

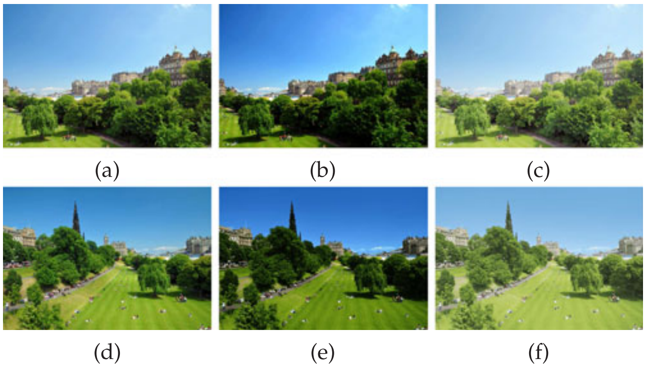


Fig. 9. Examples of weather counterpart mapping. (b) and (c) are weather counterparts of (a). We build the mapping function from (a) to (b)-(c) respectively denoted as $\mathcal{H}_1$ and $\mathcal{H}_2$. Given the testing image (d), (a) is among those visually similar to (d) in the weather counterpart dataset. (e) and (f) are respectively the mapped results of $\mathcal{H}_1$ and $\mathcal{H}_2$ with input (a).

computer vision background. To avoid editing bias, these subjects were given no specific instruction except that their editing should not change the weather label of the image. The editing operation mainly includes gamut mapping, tone mapping, sharpening, blurring, etc. For each image, four edited versions are produced.

Therefore, for each of the two cases we have five (one original + four additional) images, which are ordered differently to form 20 image pairs after permutation. Finally we obtain a total of $(1{,}000 + 1{,}000) \times 20$ image pairs, which are named *weather counterpart image pairs*.

Weather Counterpart Mapping Functions. We now learn the mapping relationship from a given image to its weather counterpart. We assume the RGB color of an outdoor image (denoted as I) and its weather counterpart image (denoted as I^o) has the mapping relationship expressed as

$$\{r^o, g^o, b^o\} = \mathcal{H}(r, g, b), \tag{7}$$

where $\{r, g, b\}$ and $\{r^o, g^o, b^o\}$ are the input and mapped RGB color vectors. Eq. (7) has only three variables, so we can build a 3D array to turn this mapping operator into a table look-up operation.

The table construction is as follows. The intensity in each variable (channel) is quantized into 256 bins, which produces the mapping table with dimension $256 \times 256 \times 256$. We define $\mathbf{v}_p$ and $\mathbf{v}_p^o$ as the quantized RGB vector of pixel color at p in I and I^o respectively. For an input (r, g, b), by defining $\mathbf{c} = [r, g, b]^T$, the output of the mapping table is

$$\mathcal{H}(r, g, b) = \frac{1}{K} \sum_{p \in \mathcal{D}(\mathbf{c})} \left\{ \exp\left[-\frac{1}{\sigma^2} \|\mathbf{v}_p - \mathbf{c}\|_2^2 \right] \mathbf{v}_p^o + \gamma \mathbf{c} \right\}, \tag{8}$$

where

$$\mathcal{D}(\mathbf{c}) = \{ q \mid \|\mathbf{v}_q - \mathbf{c}\|_2^2 \leq \delta, \forall q \in \mathbf{\Theta} \}. \tag{9}$$

Here $\mathbf{\Theta}$ is the pixel set of the image. The K in Eq. (8) is a normalization factor, which is given by

$$K = \sum_{p \in \mathcal{D}(\mathbf{c})} \exp\left[-\frac{1}{\sigma^2} \|\mathbf{v}_p - \mathbf{c}\|_2^2 \right] + \gamma. \tag{10}$$

In our experiments, we set $\delta = 45$, $\gamma = 0.1$ and $\sigma = 5$. The output of $\mathcal{H}(r, g, b)$ is the weighted combination of pixel-pairs mapping whose input RGB vector is close to $[r, g, b]^T$. Therefore, the mapping relationship of I and I^o can be smoothly transferred to table $\mathcal{H}$. If we cannot find sufficient number of RGB vectors close to $[r, g, b]^T$ in the original image, $\sum_{p \in \mathcal{D}(\mathbf{c})} \exp[-\frac{1}{\sigma^2} \|\mathbf{v}_p - \mathbf{c}\|_2^2]$ is small, so that we trust more the original input $\mathbf{c}$ in this case.

Weather Counterparts Generation. We produce 40,000 mapping functions for all 40,000 weather counterpart pairs in our dataset. Each function captures a color transform from the first image to the second in the corresponding weather counterpart pair. Given a training image, we use the mapping where the input images are similar to produce the weather counterpart images for training.

We use the color GIST descriptors [8] to extract the color and context information. We pick $d = 50$ mapping functions whose input image is closest to the given image according to color GIST features, and then use the d mapping functions to produce d weather counterpart images. Fig. 9 shows examples of weather counterpart images.

5 Collaborative Learning with Homogeneous Voters

Traditional classifiers such as SVM cannot achieve good performance on our overall weather feature because they assume all of the components are present simultaneously in every image, which may unfortunately not be the case. For example, outdoor images do not always contain the sky region. Images lacking one or more weather cues would significantly affect SVM's classification performance.

Our learning strategy is to partition the training images into disjoint clusters of homogeneous voters, so that voters closer to a given testing image have more weights when the weather label is considered.

5.1 Voting Scheme

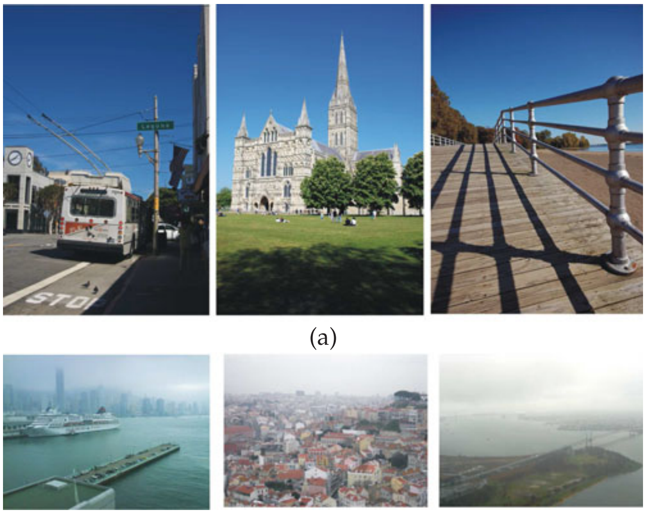
Our training outdoor images are first partitioned into homogeneous clusters according to the existence vector of each image as defined in Eq. (2). The partitioned sets thus correspond to different weather cue patterns, such as "reflection + shadow", "sky + haze", and "sky + reflection + shadow". Images in the same cluster/pattern are the homogeneous.

In implementation, we partition the set of training images into M subsets $\{\mathbf{\Omega}_1, \ldots, \mathbf{\Omega}_M\}$ based on the existence vectors using hierarchical clustering [10]. We set the cluster error threshold to 0.5 in terms of Euclidean distance. M can be found automatically. We denote the set of cluster center vectors as $\{\widehat{e}_1, \ldots, \widehat{e}_M\}$. Fig. 10 shows sample images of two converged clusters and their cluster centers.

In the testing phase, given an overall weather feature $\mathbf{x}$ with existence vector e, the training data whose existence vectors are similar to e should be used. So our classifier is implemented using a weighted voting scheme, expressed as

$$h(\mathbf{x}, e) = sign\left[\sum_{i=1}^{M} s(\widehat{e}_i, e) \widehat{h}_i(\mathbf{x}) \right], \tag{11}$$

where $sign[\cdot]$ is the function outputting 1 (resp. -1) for non-negative (resp. negative) input, $s(\widehat{e}_i, e)$ is a similarity function under parameter σ



(a) (b)

Fig. 10. Sample images found in two clusters. (a) "sky + shadow" cluster with center $\{\mathbf{0.90}, \mathbf{0.87}, 0.26, 0.11\}$. (b) "sky + haze" cluster with center $\{\mathbf{0.94}, 0.24, 0.27, \mathbf{0.84}\}$, where $\{v_{sk}, v_{sh}, v_{re}, v_{ha}\}$ is composed of the respective existence scores.

$$s(\widehat{e}_i, e) = \frac{\exp(-\frac{\|\widehat{e_i} - e\|_2^2}{2\sigma^2})}{\sum_i^M \exp(-\frac{\|\widehat{e_i} - e\|_2^2}{2\sigma^2})}, \tag{12}$$

and $\widehat{h}_i(\cdot)$ (defined shortly) is the homogeneous voter trained using the data in Ω_i. Our classifier Eq. (11) gives a larger weight to the homogeneous voter whose existence vector pattern is similar to that of the testing data.

5.2 Collaborative Learning

For training image i, we denote the overall weather feature as $\mathbf{x}_i$, and the weather label as $y_i \in \{-1, +1\}$, where -1 and $+1$ correspond respectively to "cloudy" and "sunny". For each homogeneous voter, we model $\widehat{h}_i(\cdot)$ as

$$\widehat{h}_i(\mathbf{x}) = sign\left(\sum_{j=1}^{p} \omega_{j,i}\mathbf{x}(j) + b_i\right), \tag{13}$$

where $\mathbf{x}(j)$ is the jth element of vector $\mathbf{x}$. If each homogeneous voter works independently without information sharing, the classifier in Eq. (13) can be modeled as a standard SVM [5], expressed as

$$\begin{aligned} &\min_{\omega_{j,i}, b_i, \zeta_{i,k}} \sum_{j=1}^{p} \omega_{j,i}^2 + C \sum_{k \in \Omega_i} \zeta_{i,k} \\ &\text{s.t.} \quad y_k\left(\sum_{j=1}^{p} \omega_{j,i}\mathbf{x}_k(j) + b_i\right) \geq 1 - \zeta_{i,k}, \zeta_{i,k} \geq 0, \ \forall k \in \Omega_i, \end{aligned} \tag{14}$$

where $p = 4{,}717$ is the dimension of the overall weather feature and C is a constant.

In our framework, we do not train each $\widehat{h}_i(\mathbf{x})$ independently because this will lead to a large bias. Our voters work collaboratively to determine the classification result and we optimize them together in a unified framework.

By removing $sign$ from $\widehat{h}_i(\mathbf{x})$, we make the system linear, which updates Eq. (11) into

$$h(\mathbf{x}, e) = sign\left[\sum_{i=1}^{M} s(\widehat{e}_i, e)\left(\sum_{j=1}^{p} \omega_{j,i}\mathbf{x}_k(j) + b_i\right)\right]. \tag{15}$$

We make this change because a voter should not be restricted to output binary values. This also helps to indicate ambiguous situation where sunny and cloudy features are present at the same time, see Fig. 19.

5.3 Latent SVM Learning

Now, for each training sample, we produce d weather counterpart images and extract the 4717-D weather feature on them. Given training image t, we denote the weather feature of its lth weather counterpart image as $\mathbf{x}_t^l$ $(l = \{1, \ldots, d\})$. We also define $\mathbf{x}_t^0 = \mathbf{x}_t$.

For each training sample, we require that its weather counterparts to have the same weather label in the training phase. This requires us to prevent all the weather counterpart features from falling into the margin during the training stage. To this end, we define a latent variable to indicate which weather counterpart image can produce the minimum classification margin, and encourage the minimum margin to be large during optimization. According to the max-margin strategy, we can write the constraints as

$$\begin{aligned} &\min_{c \in \{0,\ldots,d\}} \left\{ y_k\left(\sum_{j=1}^{p} \omega_{j,i}\mathbf{x}_k^c(j) + b_i\right)\right\} \geq 1 - \zeta_{i,k}, \\ &\zeta_{i,k} \geq 0, \ \forall k \in \Omega_i, \ \forall i = 1, \ldots, M, \end{aligned} \tag{16}$$

and

$$\begin{aligned} &\min_{m \in \{0,\ldots,d\}} \left\{ y_t\left[\sum_{i}^{M} s(\widehat{e}_i, e_t)\left(\sum_{j=1}^{p} \omega_{j,i}\mathbf{x}_t^m(j) + b_i\right)\right]\right\} \geq 1 - \xi_t, \\ &\xi_t \geq 0, \ \forall i = 1, \ldots, M, \ \forall t = 1, \ldots, N, \end{aligned} \tag{17}$$

where N is the number of training images, c and m are two latent valuables to indicate which weather counterpart image produces the minimum classification margin.

Denoting the latent variables as $\mathcal{C} = \{c(1), \ldots, c(N)\}$ and $\mathcal{M} = \{m(1), \ldots, m(N)\}$ for Eqs. (16) and (17) respectively, the final objective function for $h(\mathbf{x}, e)$ is written as

$$\min_{\omega_{j,i}, b_i, \xi_t, \zeta_{i,k}, \mathcal{C}, \mathcal{M}} \sum_{i=1}^{M} \sum_{j=1}^{p} \omega_{j,i}^2 + C_1 \sum_{i=1}^{M} \sum_{k \in \Omega_i} \zeta_{i,k} + C_2 \sum_{t=1}^{N} \xi_t \tag{18}$$

s.t.

$$\begin{aligned} &\min_{c(k) \in \{0,\ldots,d\}} \left\{ y_k\left(\sum_{j=1}^{p} \omega_{j,i}\mathbf{x}_k^{c(k)}(j) + b_i\right)\right\} \geq 1 - \zeta_{i,k}, \\ &\zeta_{i,k} \geq 0, \ \forall k \in \Omega_i, \ \forall i = 1, \ldots, M \end{aligned} \tag{19}$$

$$\begin{aligned} &\min_{m(t) \in \{0,\ldots,d\}} \left\{ y_t\left[\sum_{i}^{M} s(\widehat{e}_i, e_t)\left(\sum_{j=1}^{p} \omega_{j,i}\mathbf{x}_t^{m(t)}(j) + b_i\right)\right]\right\} \geq 1 - \xi_t \\ &\xi_t \geq 0, \ \forall i = 1, \ldots, M, \ \forall t = 1, \ldots, N, \end{aligned} \tag{20}$$

where C_1 and C_2 are constants.

Eq. (18) can be regarded as latent SVM whose latent variables are $\mathcal{C}$ and $\mathcal{M}$. We solve Eq. (18) by iteratively

TABLE 3
Classification Results (Mean $\pm$ Variance) Using Individual Features

Feature	Normalized accuracy
Sky	44.3 ± 1.9
Shadow	41.1 ± 2.2
Reflection	27.0 ± 2.1
Contrast	40.5 ± 2.0
Haze	34.1 ± 1.9
CNN feature	83.6 ± 2.0

optimizing $\{\omega_{j,i}, b_i, \xi_t, \zeta_{i,k}\}$ as a standard SVM problem, and optimize the latent values $\{\mathcal{C}, \mathcal{M}\}$ in the constraining conditions. The following steps are adopted:

1) Keep $\{\omega_{j,i}, b_i, \xi_t, \zeta_{i,k}\}$ fixed, optimize the latent $\{\mathcal{C}, \mathcal{M}\}$ subject to the constraints (19) and (20). This is a simple minimization operation.
2) Keep $\{\mathcal{C}, \mathcal{M}\}$ fixed, optimize $\{\omega_{j,i}, b_i, \xi_t, \zeta_{i,k}\}$ by solving a standard SVM problem which can be solved using Lagrange multipliers [5].

For the second step, voter collaboration is characterized by Eq. (20), which forces all of the voters to work together in the classification. The effectiveness of each voter is governed by Eq. (19). It guarantees that each voter is learned from its corresponding homogeneous data. Eqs. (19) and (20) can accomplish good classification performance. We solve Eq. (18) using different σs for $s(\cdot)$ in Eq. (12). In the final stage, we pick the σ with the minimum energy for the objective function (18).

Similar to other latent SVM solvers, the two-step iteration converges to a satisfactory $\{\omega_{j,i}, b_i\}$ in our experiments. This is because the difference among feature weather counterpart images is much smaller than the difference among all of the training samples. In the two-step iteration, compared to the first step (latent variables optimization), the second step plays a more important role in model update with SVM weights. The latent variable optimization can be regarded as a fine-tuning step in the feature space, in order to capture the variance introduced by different camera parameters and image editing. Empirically, the update stops after 8-10 iterations.

6 Experiments

We report the classification results under different evaluation settings, and further validate our dataset using a perceptual study. In the following, the *CNN feature* is the feature extracted from the CNN model used as input to our system. The *CNN classifier* refers to the CNN model trained end-to-end on our dataset. The visualization of the learned convolutional filters can be found in Fig. 14.

6.1 Classification Results

The training and classification were done using the weather dataset constructed. We adopted the cross validation scheme where in each round, 80 percent of the data were selected randomly as the training set, with the remaining 20 percent as the testing set. We ran five rounds of experiment and recorded the mean and variance of the classification accuracy.

On two-class labeling, even random guess can reach 50 percent accuracy. We use the normalized accuracy given by $\max\{(a-0.5)/(1-0.5), 0\}$, where a is the raw accuracy obtained. Thus, the normalized accuracy is within $[0, 1]$ and random guess is expected to get zero.

6.1.1 *Individual Features and Scores*

We use SVM to evaluate individual weather features. Note that it may not be fair for the CNN feature whose performance is sensitive to choice of parameters. Nevertheless we include the CNN feature in this section, and defer the discussion of its performance under different parameter settings in a later section. The momentum, weight decay and learning rate of the CNN are 0.0001, 0.9 and 0.005 respectively to produce the best performance.

Table 3 tabulates the classification results. Although the overall weather feature vector (4717D) is lopsided to the CNN feature (4096D), and that using the CNN feature alone reports better performance than any of the individual non-CNN features, we shall show that the combination of CNN and non-CNN features reports the best performance by taking the advantages of both features. Not surprisingly, the sky is the most important weather cue among the five non-CNN features. We believe that this is due to the fact that sky detection is relatively easier and more stable. The majority of failure cases are images without a prominent sky region. In addition, the reflection and shadow classifiers also work well. The performance of the contrast classifier on the other hand depends on the complexity of the scene.

We note that the haze cue is weaker than the sky and contrast cues mainly due to the fact that many images in our dataset simply do not exhibit detectable haze. To confirm this, we select 415 images with haze v_{ha} score larger than 0.7 and 415 sunny images. The haze classifier performance is improved up to 84.2 percent in normalized accuracy when it is applied to these 830 images. We also found that the haze cue can help identify sunny images as well in classification, since many sunny images have vivid color which exhibits low dark-channel intensities.

Next, we evaluate individual existence scores, which are used to form Eq. (2). For each individual feature, we select s percent of the images with the highest existence score in the dataset, and apply SVM classification on this image subset. Fig. 11 shows the performance with varying s of each individual classifier. The plot indicates that our existence score design is effective—each individual feature is more useful when it has a higher existence score. For reference, two fail cases produced by our system are shown in Fig. 18.

6.1.2 *Ablation Study*

We also conducted an ablation study, that is, to study the performance of the system when a given weather cue is left out. Table 4 verifies that all of the proposed cues are useful in accounting for the overall weather classification performance, since removing any one of them results in a performance drop. In particular, the momentum, weight decay and learning rate of our CNN are 0.0001, 0.9 and 0.005

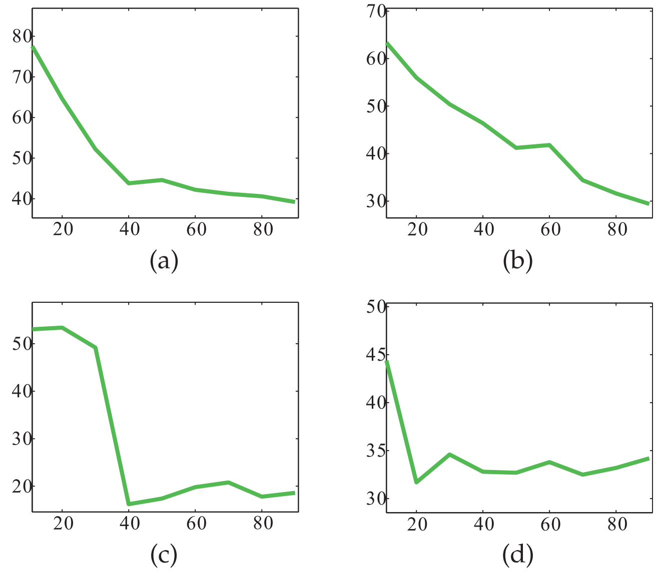


Fig. 11. (a)-(d) are respectively the performance curves of sky, shadow, reflection, and haze classifiers. The x-axis values are the respective percentages of selected images (with the highest existence score) in the dataset. The y-axis is the classification accuracy (in percent).

respectively. We note that the CNN feature plays an important role among all of the features due to the larger drop in comparison to other features.

6.1.3 Data Augmentation

We enrich the data set by introducing weather counterpart images in the training phase. In this section we compare the classification results on training using the data with and without data augmentation. Table 5 shows that our data augmentation leads to about 7 percent improvement in normalized accuracy. Note that we use the best parameters for the CNN classifier and our classifier.

Therefore, for the rest of our evaluation in this section we will use the augmented training dataset.

6.1.4 Comparison

We report our overall classification performance compared with typical baseline systems, weather related systems, and the CNN classifier. For the latter, we will show and explain that relying on the CNN without the proposed weather-specific features will result in a significant performance drop, despite that CNN is powerful in encoding global scene structure and characteristics.

TABLE 4
Classification Results (Mean $\pm$ Variance) with Individual Weather Cues Being Left Out

Feature	Normalized accuracy
Sky	86.3 ± 1.9
Shadow	86.9 ± 2.1
Reflection	89.0 ± 2.0
Contrast	86.2 ± 1.9
Haze	87.7 ± 1.5
CNN feature	61.4 ± 2.0
All	$\mathbf{91.4 \pm 1.6}$

"All" means individual weather cues are included.

TABLE 5
Classification Results (Mean $\pm$ Variance) with and without Data Augmentation; "with DA" and "without DA" Respectively Stand for Training with and without Data Augmentation

Classifier	Normalized accuracy
CNN classifier (without DA)	77.8 ± 2.0
CNN classifier (with DA)	83.3 ± 2.1
Ours (without DA)	84.0 ± 1.8
Ours (with DA)	$\mathbf{91.4 \pm 1.6}$

The learning rate, momentum and weight decay of the CNN are respectively 0.0001, 0.9 and 0.005.

Comparison with Baseline Systems. The first baseline is to implement SVM directly on the 4717-D weather feature. We test both the linear and non-linear versions with different kernels and report the results with the best performance. The second baseline is the traditional Adaboost, which combines several classifiers to build a stronger one. We take each feature bin as a weak classifier. Another two baseline methods based on dictionary learning [39] are typical image classification methods, namely LLC [58] and ScSPM [62].

For the SVM baseline we tried different parameters: $C \in [0.0001, 100]$ and we report the best one ($C = 0.05$). We tried different non-linear kernels, including the Gaussian kernel, RBF kernel and Polynomial kernel. The best kernel is the polynomial kernel which produces a performance similar to the linear solver. For the Adaboost baseline, we took each single dimension in the 1645-D feature as a weak classifier, and tried 50 different permutations of the weak classifiers. We found that the variance in accuracy is very small—0.16 only. For LLC and ScSPM, we use the codes provided in [58] and [62] with default parameters. We also tried three dictionary sizes, 512, 1,024 and 2,048. We found all of the parameters cannot yield a result significantly greater than 0 normalized accuracy.

Table 6 lists the classification results. Figs. 12 and 13 show a few examples, where we test five different σ values in Eq. (12), that is, $\{0.5, 0.1, 0.01, 0.05, 0.001\}$, and select the best result with the lowest energy in Eq. (18).

For traditional image classification methods LLC [58] and ScSPM [62], the normalized accuracies are close to 0. This is because these methods rely on scene structure and do not consider illumination information. SVM and Adaboost do not yield significant improvement over single weather classifiers, such as those of sky or shadow (cf. Tables 3 and 6). We also find that the use of kernel SVM yields similar performance.

TABLE 6
Classification Results (Mean $\pm$ Variance) of Different Methods

	Normalized accuracy
SVM	41.2 ± 2.2
Adaboost	36.4 ± 2.3
LLC [58]	0.3 ± 0.1
ScSPM [62]	0.2 ± 0.1
CNN classifier [22]	83.3 ± 1.8
Ours	$\mathbf{91.4 \pm 1.6}$

The momentum, weight decay and learning rate of the CNN are 0.0001, 0.9 and 0.005 respectively.

Fig. 12. Detection results: Cloudy images.

Fig. 13. Detection results: Sunny images.

Comparison with Related Methods. We also compare our classifier with weather-related methods. The first related work is Lalonde et al. [26]. Note that their system is not designed for weather classification; one component, namely, the sun visibility prediction, can be regarded as a coarse weather estimator. We implemented this component and tested it on our dataset. Another two vehicle-based weather classifiers [46], [61] were also compared. The comparison with [24] is also provided. It can output 40 attributes of image feature including "sunny" and "cloudy". We train the regressor on our data.

Table 7 tabulates the classification statistics. For the method of [26], the assumption that an outdoor scene is composed of ground, sky, and vertical surfaces may not be satisfied (see a few exceptions in Figs. 12 and 13). For the work of [46], [61], the weather estimators are specially designed for driver assistance. They rely on vehicle-mounted image priors, which cannot properly deal with general natural images.

The framework proposed in [24], where off-the-shelf but not weather-specific features are used, produces less effective results. But it still outperforms [46], [61] since it does not rely on on vehicle-mounted priors. We also observe that the precisions for "cloudy" and "sunny" attributes reported on the dataset of [24] is high (> 0.95), because the testing images in [24] are dominated by a prominent sky region which is relatively easy to recognize.

Comparison with CNN. In this section, we train a CNN classifier end-to-end to perform the two-class classification. The standard structure provided by [22] is adopted, and we fine tune on the AlexNet. We report the result with the best parameter setting.

For the sake of fairness, the CNN feature in our classifier adopts the same set of parameters with the one being compared. The results are reported with various hyperparameters.

First, in Table 8 we show the performance under different learning rates; the most important parameter for CNN training. The other two important parameters are namely the momentum and weight decay. Table 9 summarizes their effects in the comparison experiments. We found that the proposed non-CNN weather features are complementary with the CNN feature in the overall weather feature vector. That is, while the CNN feature is capable of capturing global image characteristics, it may not encode well weather characteristics which are better represented by non-CNN weather cues. The result shows that our system which combines CNN feature and non-CNN features leads to about 8 percent improvement.

To further verify this point, we manually label non-CNN weather cues to explore their full potentials. For the shadow feature, we manually label shadow regions in all of the images using bounding boxes, followed by shadow extraction [25] within the box region. The existence score is 1 (with shadow) or 0 (without shadow). For the reflection feature, we label reflection regions manually with a one-dimensional feature. If reflection occurs the feature bin is 1; otherwise 0. The existence score is the same as the feature bin value. For the sky feature, with a few exceptions most of the sky regions can be correctly segmented, and we manually label the missing sky regions. Table 10 tabulates the results, which demonstrates that the proposed features are indeed effective in weather recognition and perform better

TABLE 7
Classification Statistics of Different Methods

	Normalized accuracy
Lalonde et al. [26]	46.5 ± 1.7
Yan et al. [61]	24.6 ± 2.6
Roser and Moosmann [46]	26.2 ± 2.3
Laffont et al. [24]	21.4 ± 1.9
Ours	$\mathbf{91.4 \pm 1.6}$

TABLE 8
Comparison with CNN Classifier under Different Learning Rates, Given Momentum and Weight Decay Are Respectively 0.9 and 0.005

learning rate	CNN classifier	Ours
0.001	81.5 ± 1.9	88.7 ± 2.0
0.0001	83.3 ± 1.8	$\mathbf{91.4 \pm 1.6}$
0.00001	82.0 ± 2.0	89.5 ± 1.9

TABLE 9
Comparison with CNN Classifier, Given Learning Rate Is 0.0001, ν and ξ Are Respectively Momentum and Weight Decay

parameters	CNN classifier	Ours
$\mathbf{\nu = 0.9, \xi = 0.005}$	83.3 ± 1.8	$\mathbf{91.4 \pm 1.6}$
$\nu = 0.9, \xi = 0.0025$	81.4 ± 1.8	89.6 ± 1.7
$\nu = 0.45, \xi = 0.005$	77.1 ± 2.1	85.2 ± 1.8
$\nu = 0.45, \xi = 0.0025$	78.6 ± 1.9	86.7 ± 2.0

TABLE 10
Comparison with Manual Feature Localization

	Normalized accuracy
CNN classifier	83.3 ± 1.8
CNN classifier by [9]	82.2 ± 3.5
Ours (Sky)	93.5 ± 1.8
Ours (Shadow)	96.3 ± 1.7
Ours (Reflection)	95.2 ± 1.6
Ours (Sky + Shadow + Reflection)	$\mathbf{97.4 \pm 1.2}$

In the table, the feature inside the parentheses are manually labeled as described in the text. The learning rate, momentum and weight decay for the CNN classifier are respectively 0.0001, 0.9 and 0.005.

than the CNN classifier. Thus, we believe that with the continuing improvement of low-level vision techniques, the proposed non-CNN weather cues will improve the overall performance significantly by working in synergy with the

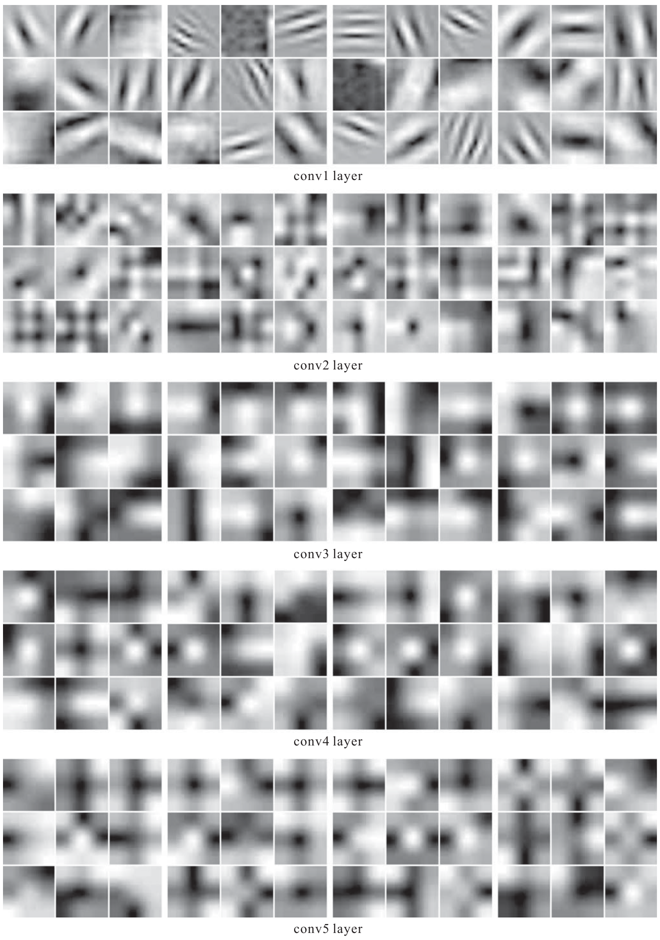


Fig. 14. The visualization of the filters of conv1-conv5 layers.

sunny 0.02 cloudy 0.98 (a)	sunny 0.03 cloudy 0.97 (b)	sunny 0.97 cloudy 0.03 (c)	sunny 0.94 cloudy 0.06 (d)

Fig. 15. Four CNN failure cases. In (a) and (b), Sunny image mis-detected as cloudy. In (c) and (d) cloudy image mis-detected as sunny. “sunny score” and “cloudy score” refer to output CNN confidence score in the sunny and cloudy class.

CNN feature. Fig. 15 shows some fail cases of the CNN classifier which can otherwise be correctly labeled by our method. For example, Figs. 15a and 15b are recognized as cloudy by the CNN classifier due to its globally gray color tone. But our system can look into more details such as shadows to produce the correct weather label. In Figs. 15c and 15d, although shadows are found, they are not strong cast shadow caused by the sun. On the other hand, the small sky region can be correctly detected by our method.

We visualize the CNN filters in different layers as shown in Fig. 14.

6.2 Comparison on Two-Class Laffont Dataset

In [24], a dataset including 40 transient attributes was proposed. In this dataset, 8,571 images in total from 101 webcams are annotated by crowd-sourcing. “Sunny” and “cloudy” are two of the attributes in their setting. That is, for each image, confidence score of “sunny” and “cloudy” are available. As suggested in [24], attributes with confidence score larger than 0.8 is considered strong positive attributes. The images with strong positive sunny/cloudy attributes are selected to form the two-class Laffont dataset. We found in this dataset none of the images is labeled both “sunny” and “cloudy”, so the images are unambiguous. The resulting dataset contains 1,729 cloudy images and 1,085 sunny images. Fig. 16 shows example images of the *two-class Laffont dataset*. We apply different methods on the dataset by using 80 percent of it for training and 20 percent for testing, and report the cross-validation results. Table 11 shows that our method has a better performance than [24] on the two-class dataset. In particular, note that the performance of our method and the CNN classifier are close to 100 percent. This is because the images in the dataset typically have a prominant sky region which makes the classification easier.

6.3 Application in Weather Monitoring Using Surveillance Cameras

We verify our technique in a real-world application: real-time weather monitoring. Surveillance cameras can be found almost everywhere, so we believe running our fast and cost-effective method on these cameras can effectively

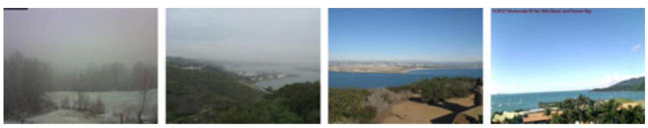


Fig. 16. Examples of the two-class Laffont dataset.

TABLE 11
Comparison on the Two-Class Laffont Dataset

	Normalized Accuracy
[24]	92.4 ± 1.3
CNN classifier	96.4 ± 0.5
Ours without CNN feature	98.2 ± 0.1
Ours	$\mathbf{98.6 \pm 0.2}$

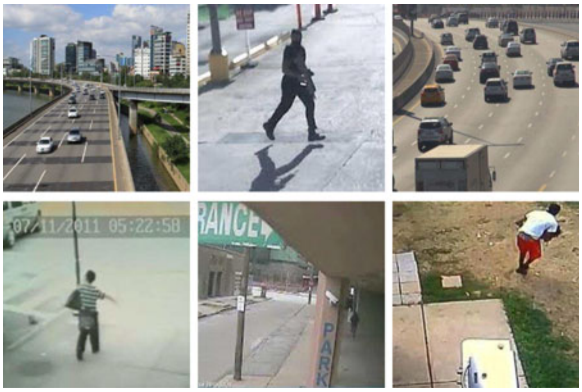


Fig. 17. Some detection results for surveillance images. The first and second row are sunny and cloudy images respectively.

monitor real-time weather in urban areas. This is particularly useful where solar panels are extensively installed on the rooftops of many buildings; sunny and cloudy weather provides important guidance for optimizing power transfer in the main grids. That is, when cloudy weather is detected for an extended period of time, the main power grid may start to take over early to maintain stable power supply and avoid outage. We apply our weather predictor in surveillance videos footage, and collected 2,000 surveillance images, 1,000 of them are sunny and the other 1,000 are cloudy. The normalized classification accuracy is 93.2 percent. Sample images are shown in Fig. 17.

7 Conclusion and Future Work

We have presented a learning-based approach for classifying two types of weather. This apparently simple two-class weather labeling problem is not trivial given the great variety of outdoor images. The feature cues we used resonate well with our own common sense in judging weather conditions. Because some of the feature cues may be unavailable in images, the key to our computational framework is a collaborative learning strategy where voters closer to the testing image in terms of weather information/structure are given more weight in classification. We have also incorporated the powerful CNN feature into our overall weather feature. To resist variations caused by different camera parameters and photo editing, a latent SVM framework is proposed to learn from various synthesized weather counterpart images. Our experimental results showed that this is an effective strategy, which we believe has good potential beyond weather classification.

Our current approach is limited to label two weather types. More research needs to be engaged in generalizing the approach to labeling more conditions on larger dataset [4]. For example, Fig. 19 shows two images where sunny and cloudy features are present at the same time. They may be

(a) (b)

Fig. 18. (a) Sunny image mis-detected as cloudy, and (b) cloudy image mis-detected as sunny.

(a) (b)

Fig. 19. Sunny or cloudy?

labeled as "partly sunny" or "partly cloudy" and in fact, our system labels (a) as sunny, with the rescaled SVM sunny score 0.641 (and cloudy score 0.359), while labeling (b) as cloudy with the rescaled SVM cloudy score 0.716 (and sunny score 0.284), which we believe are reasonable for two-class weather classification.

We hope this paper will spark interest and subsequent work along this line of research. Executable and the weather dataset are available at the project website.

Acknowledgments

The research was supported by the Research Grants Council of the Hong Kong Special Administrative Region (Project nos. 413113, 619313 and 412911). It was also partly supported by the National Natural Science Foundation of China (NSFC) key project No. 61133009. Di Lin is the corresponding author of this paper.

References

[1] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *Proc. 2nd Int. Conf. Human Behavior Understanding*, 2011, pp. 29–39.
[2] R. Baltenberger, M. Zhai, C. Greenwell, S. Workman, and N. Jacobs, "A fast method for estimating transient scene attributes," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2016, pp. 1–8.
[3] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern*, 2010, pp. 2559–2566.
[4] W. T. Chu, X. Y. Zheng, and D. S. Ding, "Image2Weather: A large-scale image dataset for weather property estimation," in *Proc. IEEE 2nd Int. Conf. Multimedia Big Data*, Apr. 2016, pp. 137–144.
[5] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, pp. 273–297, 1995.
[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 886–893.
[7] K. Derpanis, M. Lecce, K. Daniilidis, and R. Wildes, "Dynamic scene understanding: The role of orientation features in space and time in scene classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1306–1313.
[8] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid, "Evaluation of GIST descriptors for web-scale image search," in *Proc. ACM Int. Conf. Image Video Retrieval*, 2009, Art. no. 19.
[9] M. Elhoseiny, S. Huang, and A. Elgammal, "Weather classification with deep convolutional neural networks," in *Proc. IEEE Int. Conf. Image Process.*, 2015, pp. 3349–3353.

[10] M. Fionn, "A survey of recent advances in hierarchical clustering algorithms," *Comput. J.*, vol. 26, pp. 354–359, 1983.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[12] D. Glasner, P. Fua, T. Zickler, and L. Zelnik-Manor, "Hot or not: Exploring correlations between appearance and temperature," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3997–4005.

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

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conf. Comput. Vis. Patt. Recog.*, Las Vegas, NV, USA, pp. 770–778, Jun. 2016, Doi: 10.1109/CVPR.2016.90.

[15] M. Islam, N. Jacobs, H. Wu, and R. Souvenir, "Images+ weather: Collection, validation, and refinement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop Ground Truth*, vol. 6, p. 2, 2013.

[16] N. Jacobs, M. T. Islam, and S. Workman, "Cloud motion as a calibration cue," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1344–1351.

[17] M. Juneja, A. Vedaldi, C. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 923–930.

[18] H. Kang, M. Hebert, and T. Kanade, "Discovering object instances from scenes of daily living," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 762–769.

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1725–1732.

[20] H. Katsura, J. Miura, M. Hild, and Y. Shirai, "A view-based outdoor navigation using object recognition robust to changes of weather and seasons," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, pp. 2974–2979.

[21] G. Kim and A. Torralba, "Unsupervised detection of regions of interest using iterative link analysis," in *Proc. 22nd Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 961–969.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Advances Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[23] H. Kurihata, et al., "Rainy weather recognition from in-vehicle camera images for driver assistance," in *Proc. IEEE Intell. Veh. Symp.*, 2005, pp. 205–210.

[24] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays, "Transient attributes for high-level understanding and editing of outdoor scenes," *ACM Trans. Graph.*, vol. 33, no. 4, 2014, Art. no. 149.

[25] J.-F. Lalonde, A. Efros, and S. Narasimhan, "Detecting ground shadows in outdoor consumer photographs," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 322–335.

[26] J.-F. Lalonde, A. Efros, and S. Narasimhan, "Estimating the natural illumiation conditions from a single outdoor image," *Int. J. Comput. Vis.*, vol. 98, pp. 123–145, 2012.

[27] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 2169–2178.

[28] Y. J. Lee and K. Grauman, "Object-graphs for context-aware category discovery," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1–8.

[29] A. Levin, D. Lischinski, and Y. Weiss, "A closed form solution to natural image matting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 61–68.

[30] L.-J. Li, H. Su, L. Fei-Fei, and E. P. Xing, "Object bank: A high-level image representation for scene classification & semantic feature sparsification," in *Proc. Advances Neural Inf. Process. Syst. 23*, 2010, pp. 1378–1386.

[31] L.-J. Li, H. Su, Y. Lim, and L. Fei-Fei, "Objects as attributes for scene classification," in *Proc. 11th Eur. Conf. Trends Topics Comput. Vis.*, 2012, pp. 57–69.

[32] Q. Li, J. Wu, and Z. Tu, "Harvesting mid-level visual concepts from large-scale internet images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 851–858.

[33] D. Lin, C. Lu, R. Liao, and J. Jia, "Learning important spatial pooling regions for scene classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3726–3733.

[34] D. Lin, X. Shen, C. Lu, and J. Jia, "Deep LAC: Deep localization, alignment and classification for fine-grained recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1666–1674.

[35] M. Lin, Q. Chen, and S. Yan, "Network in network," *Int. Conf. Learn. Represent.*, 2014.

[36] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, pp. 91–110, 2004.

[37] C. Lu, D. Lin, J. Jia, and C.-K. Tang, "Two-class weather classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3718–3725.

[38] C. Lu, J. Shi, and J. Jia, "Online robust dictionary learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 415–422.

[39] C. Lu, J. Shi, and J. Jia, "Scale adaptive dictionary learning," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 837–847, Feb. 2014.

[40] S. G. Narasimhan and S. K. Nayar, "Shedding light on the weather," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2003, pp. I-665–I-672.

[41] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, pp. 145–175, 2001.

[42] M. Pandey and S. Lazebnik, "Scene recognition and weakly supervised object localization with deformable part-based models," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 1307–1314.

[43] S. Parizi, J. Oberlin, and P. Felzenszwalb, "Reconfigurable models for scene recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2775–2782.

[44] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, "Large-scale image retrieval with compressed Fisher vectors," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3384–3391.

[45] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 413–420.

[46] M. Roser and F. Moosmann, "Classification of weather situations on single color images," in *Proc. IEEE Intell. Veh. Symp.*, 2008, pp. 798–803.

[47] B. Russell, A. Torralba, K. Murphy, and W. T. Freeman, "LabelMe: A database and web-based tool for image annotation," *Int. J. Comput. Vis.*, vol. 77, pp. 157–173, 2008.

[48] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 1605–1614.

[49] F. Sadeghi and M. F. Tappen, "Latent pyramidal regions for recognizing scenes," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 228–241.

[50] L. Shen and P. Tan, "Photometric stereo and weather estimation using internet images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1850–1857.

[51] N. Shroff, P. Turaga, and R. Chellappa, "Moving vistas: Exploiting motion for describing scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1911–1918.

[52] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, abs/1409.1556, 2014.

[53] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 73–86.

[54] L. Tao, L. Yuan, and J. Sun, "SkyFinder: Attribute-based sky image search," in *Proc. ACM SIGGRAPH*, 2009, Art. no. 68.

[55] S. Todorovic and N. Ahuja, "Unsupervised category modeling, recognition, and segmentation in images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 12, pp. 2158–2174, Dec. 2008.

[56] A. Volokitin, R. Timofte, L. Van Gool, and D. CVL, "Deep features or not: Temperature and time prediction in outdoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 63–71.

[57] J. Wang, M. Korayem, and D. Crandall, "Observing the natural world with Flickr," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2013, pp. 452–459.

[58] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3360–3367.

[59] S. Workman, R. Souvenir, and N. Jacobs, "Scene shape estimation from multiple partly cloudy days," *Comput. Vis. Image Understanding*, vol. 134, pp. 116–129, 2015.

[60] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3485–3492.

[61] X. Yan, Y. Luo, and X. Zheng, "Weather recognition based on images captured by vision system in vehicle," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 5553, pp. 390–398, 2009.

[62] 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.

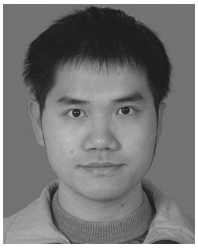
[63] B. Yao, G. Bradski, and L. Fei-Fei, "A codebook-free and annotation-free approach for fine-grained image categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3466–3473.

[64] N. Zhang, E. Shelhamer, Y. Gao, and T. Darrell, "Fine-grained pose prediction, normalization, and recognition," *CoRR*, abs/1511.07063, 2015.

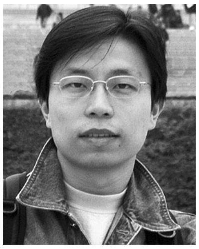
[65] Z. Zhang and H. Ma, "Multi-class weather classification on single images," in *Proc. IEEE Int. Conf. Image Process.*, 2015, pp. 4396–4400.

[66] Y. Zheng, Y.-G. Jiang, and X. Xue, "Learning hybrid part filters for scene recognition," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 172–185.

Cewu Lu received the BS and MS degrees from Chongqing University of Posts and Telecommunications and Graduate University of Chinese Academy of Sciences, in 2006 and 2009, respectively. He received the PhD degree in computer science and engineering from the Chinese University of Hong Kong, in 2013. He was a research fellow with the Hong Kong University of Science and Technology. He was a research fellow with Stanford University AI lab. Since 2016, he has been in the Department of Computer Science and Engineering, Shanghai Jiaotong University where he is currently a professor. His team received the fourth place in ILSVRC 2014 among the 38 participating teams. He received the best paper award of NPAR 2012 and served as an associate editor of the journal *Gate to Computer Vision and Pattern Recognition* and reviewers of several major computer vision and graphics conferences and journals such as the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and the *ACM Transactions on Graphics*. His research interests include activity recognition, object detection, and image/video processing. He is a member of the IEEE.



Di Lin received the bachelor's degree in software engineering from Sun Yat-sen University, in 2012, and the PhD degree from the Chinese University of Hong Kong, in 2016. His research interests include computer vision and machine learning. He is currently an assistant professor in the College of Computer Science and Software Engineering, Shenzhen University.



Jiaya Jia received the PhD degree in computer science from the Hong Kong University of Science and Technology, in 2004. He is currently a full professor in the Department of Computer Science and Engineering, Chinese University of Hong Kong (CUHK). He was a visiting scholar with Microsoft Research Asia from March 2004 to August 2005 and conducted collaborative research at Adobe Systems in 2007. He heads the research group in CUHK, focusing specifically on computational photography, 3D reconstruction, practical optimization, and motion estimation. He currently serves as an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and served as an area chair of ICCV 2011, ICCV 2013, and CVPR 2016. He is a senior member of the IEEE.

Chi-Keung Tang received the MSc and PhD degrees in computer science from the University of Southern California, Los Angeles, in 1999 and 2000, respectively. Since 2000, he has been in the Department of Computer Science, Hong Kong University of Science and Technology where he is currently a full professor. He was an adjunct researcher in the Visual Computing Group, Microsoft Research Asia. His research areas are computer vision, computer graphics, and human-computer interaction. He was an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, and was on the editorial board of the *International Journal of Computer Vision*. He served as an area chair of ICCV 2007, ICCV 2009, ICCV 2011, ICCV 2015, and as a technical papers committee member of the inaugural SIGGRAPH Asia 2008, SIGGRAPH 2011, SIGGRAPH Asia 2011, SIGGRAPH 2012, SIGGRAPH Asia 2014, and SIGGRAPH Asia 2015. He is a senior member of the IEEE.