Two-Class Weather Classification*

Cewu Lu§ Di Lin† Jiaya Jia† Chi-Keung Tang§

§The Hong Kong University of Science and Technology

†The Chinese University of Hong Kong

http://www.cse.cuhk.edu.hk/leojia/projects/weatherclassify/

Abstract

Given a single outdoor image, this paper proposes a collaborative learning approach for labeling it as either sunny or cloudy. Never adequately addressed, this two-class classification problem is by no means trivial given the great variety of outdoor images. Our weather feature combines special cues after properly encoding them into feature vectors. They then work collaboratively in synergy under a unified optimization framework that is aware of the presence (or absence) of a given weather cue during learning and classification. Extensive experiments and comparisons are performed to verify our method. We build a new weather image dataset consisting of 10K sunny and cloudy images, which is available online together with the executable.

1. Introduction

Rain and sunshine make up our everyday weather experience. The weather affects our daily lives in many ways, from solar technologies, outdoor sporting events, to the sort of clothes we wear and whether to stay indoors or not on weekend.

While current accurate weather detection technologies rely on expensive sensors, for centuries weather observing tools consisted of the human eye (and various human senses as well). If we can exploit existing surveillance cameras, which are found almost everywhere, it may be possible to turn weather observing and detection into a powerful and cost-effective computer vision application. In fact, robotic vision has been benefited with better weather understanding [7].

Despite its remarkable value, weather understanding from a single image has not been thoroughly studied. While [16, 21] propose weather recognition from vehicles, they rely on priors specific to vehicles. Scene understanding approaches [4, 23] rely on structural information for categorizing the scene into different classes. The structural information is based on illumination-invariant features such as SIFT or HOG [2, 5, 15, 11]. Weather cues are more complicated and not scene specific, making conventional scene classification methods inapplicable. Unsupervised visual learning methods [12] are also unsuitable for this task owing to the lack of weather prior.

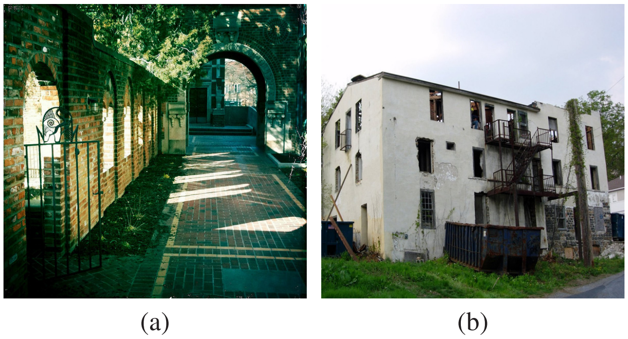


Figure 1. (a) A sunny image with mean lightness 32.41. (b) A cloudy image with mean lightness 58.25.

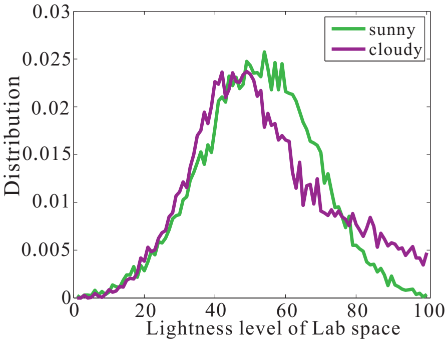


Figure 2. Pixel intensity distributions in the lightness L channel in the LAB color space of 5K cloudy images and 5K sunny images. It is almost impossible to draw a decision boundary between the two types of weather.

This paper makes the first attempt to address the two-class weather classification problem from a single outdoor image. This seemingly easy task for human – to tell whether it is a sunny or cloudy image – turns out to be challenging. Outdoor images shot during daytime can be captured at

*This research is supported by the Research Grant Council of the Hong Kong Special Administrative Region under grant numbers 412911 and 619313, and the Google Faculty Award.

different times of the day using different cameras under different weather conditions. Naive schemes based on image brightness (Figure 1) or color/intensity statistics (Figure 2) are doomed to fail in this two-class classification problem.

Our first technical contribution consists of the design and implementation of the various weather cues which are used to form the *weather feature*. These everyday weather cues (such as sky, shadow, reflection, contrast and haze) are what human are still using for weather observing – a hazy or grayish sky characterizes a cloudy day, hard shadow cast on ground indicates a sunny day, as illustrated in Figure 3(a). Conversely, in the absence of any weather cues, we ourselves would lower the confidence to correctly label weather, as shown in Figure 3(b).

Given the weather feature, the next question is how to properly learn the classifier. The main issue is that not all of the weather cues are 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 problem, our second technical contribution consists of a collaborative learning framework using homogeneous voters: we group outdoor images into clusters where images in the same cluster are similar in terms of the weather cues. This allows us to build classifiers 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 testing image is given a larger weight. As will be explained in the following, homogeneous voters are learned together in synergy under a unified optimization framework.

Despite the absence of representative work on image-based weather labeling, we perform quantitative comparison with a few common baselines including SVM, Adaboost [19, 22], and weather-related prior methods [9, 21, 16]. Perspectives on related work will be put into context when they are described.

Our final contribution consists of a 10K weather image dataset properly selected and annotated. This is used to evaluate our learning and labeling strategy.

2. The Weather Feature

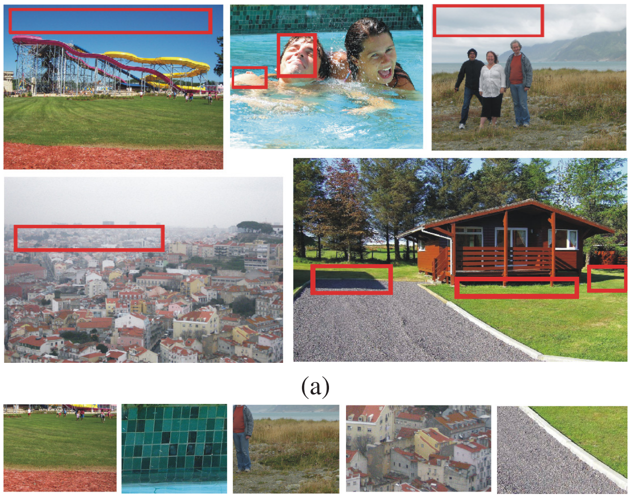
We compute for each image the *weather feature*, a 621-D feature vector formed by concatenating five components

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

where each of them, namely, sky, shadow, reflection, contrast and haze, corresponds to a key weather cue to be defined shortly. Since not all of these cues are necessarily present in a given outdoor image, we also compute the *existence vector*

$$[\ v_{sk}\ \ v_{sh}\ \ v_{re}\ \ v_{ha}\ ]^T \tag{2}$$

where each of the scalar score in $[0, 1]$ indicates the confidence in the corresponding weather cue being present in the given image. Since image contrast is present in both sunny and cloudy photos, v_{co} is always 1, and thus is not included.



(a)

(b)

Figure 3. Weather cues. (a) General weather cues in red rectangles. (b) Regions in (a) lacking any weather cues.

2.1. Sky

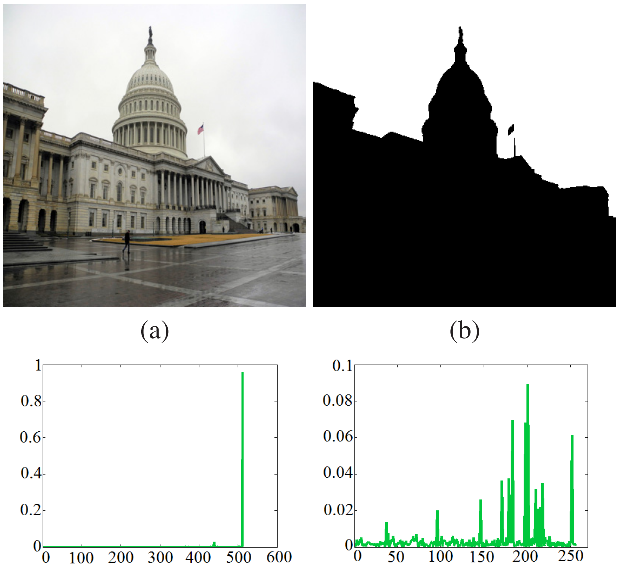
If present in an outdoor image, the sky is the most important cue for weather labeling. A clear, cloudless sky is blue as air molecules scatter blue light more than red light. Cloud is made of tiny water droplets, making sky look grey or white.

To define v_{sk}, the sky region is detected in a pixel-wise manner in the following steps. We respectively collect 20000 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 is suggested in [18]. Then random forest classifier is learned on the two patch classes. Now, given an image, we uniformly sample 15×15 patches and test their label (sky or non-sky patch) as seeds. Sky region can be segmented by implementing graph cuts on those seeds (see Figure 4(a)–(b)). Let A be the sky to image area ratio. We set $v_{sk} \in [0, 1]$ as

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

To define the $\mathbf{f}_{sk}$ vector we consider various alternatives. Straight-forward color histogram feature in the sky region suffers from two defects. First, possible sky color (both cloudy and sunny) is sparse, thus yielding most color bins with value zeros (Figure 4(c)). Second, color contrast is not considered. In this paper, we define $\mathbf{f}_{sk}$ using color-pair dictionary coding as follows.

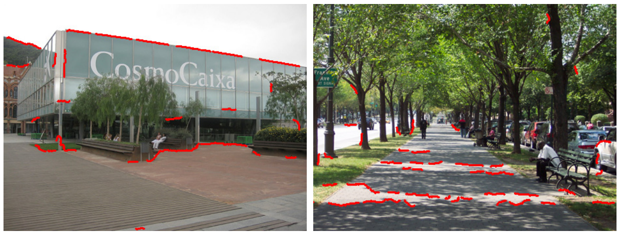
We collect 2000 images with detected sky regions. Neighborhood pixels in pairs are extracted from the sky region to form a large number of 6D vectors, each consisting of a total of 6 RGB values. This process results in about



(a) (b)

(c) (d)

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



(a) (b)

Figure 5. Shadow detection results of [8] for (a) a cloudy image and (b) a sunny image. Shadow detection in cloudy images is vulnerable to false detection.

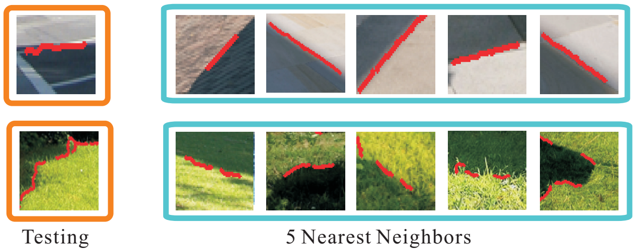


Figure 6. K-nearest neighbor matching in $\mathcal{P}$. What are shown in the blue rectangles are the five nearest neighbors.

100K pixel pairs. We then learn a sky color-pair dictionary $\mathbf{D} \in \mathbb{R}^{6\times256}$ on the vectors using the method of [13]. This results in neighborhood-pixel vectors sparsely coded over the learned dictionary, expressed as

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

where $p_i \in \mathbb{R}^{6\times1}$ is i^{th} vector, $\beta_i \in \mathbb{R}^{256\times1}$ is the sparse code over $\mathbf{D}$. We solve Eq. (4) using [19]. Our final $\mathbf{f}_{sk}$ is filtered by max pooling of all β_i. That is, the j^{th} bin of our feature set to $\max_i\{\beta_{i,j}\}$ where $\beta_{i,j}$ is the j^{th} bin of β_i.

Max pooling can preserve minor sun-to-cloud contrast in feature representation. Figure 4(d) shows a typical $\mathbf{f}_{sk}$ plot. In comparison to color histogram, our 256-D $\mathbf{f}_{sk}$ exploits the full range of the histogram and encodes color contrast information as well. We will demonstrate its advantage over color histogram in our experiments.

2.2. Shadow

Hard shadow boundaries present another useful cue as 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 detecting sky in an image, shadow detection is still a challenging problem. Our extensive evaluation indicates that 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 Figure 5.

Notwithstanding, to set v_{sh} we apply the shadow detection tool [8], rank the resulting shadow boundary confidence scores and take the 10^{th} highest score. A large v_{sh} indicates strong shadow presence.

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

In detail, initially, for all of the sunny images in the training set, we use the method of [8] 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 $10V$ samples, where V is the number of training sunny images.

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 Figure 6. The distance we use in the nearest neighbor matching is Euclidean distance of two boundaries descriptor vector provided by [8]. Given an image we obtain its top 10 most confident shadow boundaries and compute their likelihood as described above to form the 10-D $\mathbf{f}_{sh}$ vector for the image.

2.3. Reflection

Strong sunlight reflected from shiny objects is another powerful cue. Except for a perfect mirror reflector at the right reflection angle, 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 is shown in Figure 7, which compares strong sunlight reflection with the reflection from a brightly-lit matte/dull object.

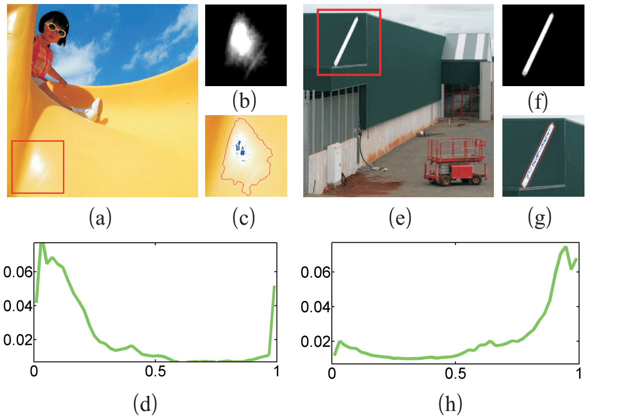


Figure 7. Reflection cue. A sunny image with strong sunlight reflection in (a) versus a cloudy image with inherently white regions in (e). (b) and (f) are the corresponding alpha mattes. In (c) and (g), red and blur points indicate background and foreground seeds used in alpha matting. (d) and (h) are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

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 [10] at the detected white pixels. The definite foreground consists of white pixels, and definite background consists of a closed curve enclosing those foreground seeds. We then estimate the closed curve under the constraint that the distance between pixels along the curve and enclosed foreground seeds should be larger than a threshold (set to 0.5 in our experiments). This closed curve can be computed by simple dynamic programming. An example is shown in Figure 7(b)–(c).

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

2.4. Contrast

Outdoor images captured in sunny and cloudy days exhibit different global and local saturation contrast, as shown in Figure 8. To compute $\mathbf{f}_{co}$, we utilize contrast information encoded as image saturation percentile. For example, a value at the 20th saturation percentile means that 20% of the image pixels are grayer than it. 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 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 i^{th} percentile in the input image 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 Figure 8(c).

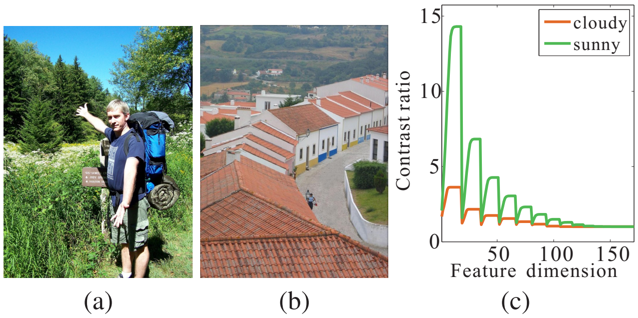


Figure 8. Contrast difference in sunny and cloudy images. (a) Sunny image, (b) cloudy image, (c) their respective contrast ratio plots, or the $\mathbf{f}_{co}$ vector.

2.5. Haze

Cloudy weather may come with haze. Haze priors have been well studied in computer vision: the dark channel prior presented in [6] is effective. Similarly, we compute the dark channel as

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

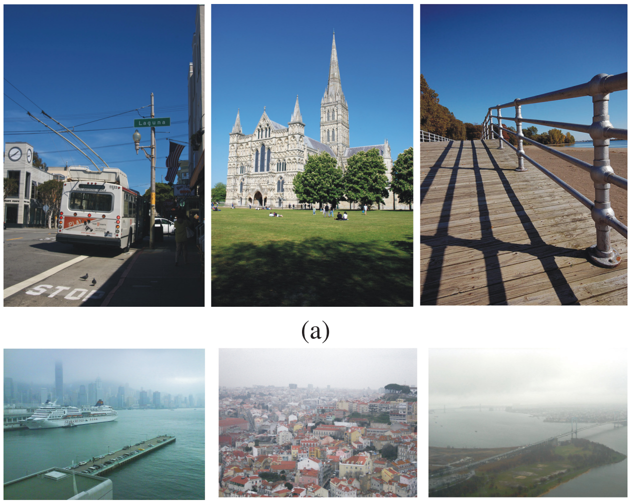
where $\mathcal{J}^c$ is a color channel and $\Omega(x)$ is a local patch (with 8×8) centered at x. Most haze-free regions have a low intensity in the dark channel. We measure the haze level and set v_{ha} of a given image as the median value of its dark channel.

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 a spatial pyramid scheme. As before, 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 dark channel intensities in these regions to form the 84-D $\mathbf{f}_{ha}$ vector.

3. Collaborative Learning with Homogeneous Voters

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

Our learning strategy is to partition training images into disjoint clusters of homogeneous voters. Given a test image, voters closer to it are given more weights for correctly finding the weather label.



(a)

(b)

Figure 9. Sample images found in two clusters of existence vector patterns. (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 respective existence scores.

3.1. 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 belonging to the same cluster/pattern are said to be homogeneous.

In implementation, we partition the set of training images into M subsets $\{\Omega_1, \ldots, \Omega_M\}$ based on existence vectors using hierarchical clustering [3]. 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 $\{\widehat{e}_1, \ldots, \widehat{e}_M\}$. Figure 9 shows sample images of two converged clusters and their cluster centers.

In the testing phase, given a weather feature $\mathbf{x}$ with existence vector e, we rely on training data whose existence vectors are similar to e. Following this idea, our classifier is implemented using a weighted voting scheme, expressed as

$$h(\mathbf{x}, e) = sign[\sum_{i=1}^{M} s(\widehat{e}_i, e)\widehat{h}_i(\mathbf{x})], \tag{6}$$

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:

$$s(\widehat{e}_i, e) = \frac{\exp(-\frac{\|\widehat{e}_i - e\|_2^2}{2\sigma^2})}{\sum_i^M \exp(-\frac{\|\widehat{e}_i - e\|_2^2}{2\sigma^2})}, \tag{7}$$

and $\widehat{h}_i(\cdot)$ (defined shortly) is the homogeneous voter trained using the data in Ω_i. Our classifier Eq. (6) gives a larger weight to the homogeneous voter whose existence vector pattern is similar to that of the testing data.

3.2. Collaborative Learning

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

$$\widehat{h}_i(\mathbf{x}) = sign(\sum_{j=1} \omega_{j,i}\mathbf{x}(j) + b_i), \tag{8}$$

where $\mathbf{x}(j)$ is the j^{th} element of vector $\mathbf{x}$. If each homogeneous voter works independently without information sharing, the classifier in Eq. (8) can be modeled as a standard SVM [1], expressed as

$$\begin{aligned} \min_{\omega_{j,i}, b_i, \zeta_{i,k}} & \sum_{j=1}^{p} \omega_{j,i}^2 + C \sum_{k \in \Omega_i} \zeta_{i,k} \\ \text{s.t. } & y_k(\sum_{j=1}^{p} \omega_{j,i}\mathbf{x}_k(j) + b_i) \geq 1 - \zeta_{i,k}, \\ & \zeta_{i,k} \geq 0, \ \forall\, k \in \Omega_i, \end{aligned} \tag{9}$$

where $p = 621$ is the dimension of weather features and C is a constant.

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

By removing *sign* from $\widehat{h}_i(\mathbf{x})$, we make the system linear, which updates Eq. (6) into

$$h(\mathbf{x}, e) = sign[\sum_{i=1}^{M} s(\widehat{e}_i, e)(\sum_{j=1}^{p} \omega_{j,i}\mathbf{x}_k(j) + b_i)]. \tag{10}$$

This change is reasonable because a voter should not be restricted to output binary values.

We follow the soft margin method in SVM to learn $h(\mathbf{x}, e)$, and rewrite the model as

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

$$\text{s.t. } y_k(\sum_{j=1}^{p} \omega_{j,i}\mathbf{x}_k(j) + b_i) \geq 1 - \zeta_{i,k}, \tag{11}$$

$$\zeta_{i,k} \geq 0, \ \forall\, k \in \Omega_i, \ \forall i = 1, \ldots, M$$

$$y_t[\sum_{i}^{M} s(\widehat{e}_i, e_t)(\sum_{j=1}^{p} \omega_{j,i}\mathbf{x}_t(j) + b_i)] \geq 1 - \xi_t \tag{12}$$

$$\xi_t \geq 0, \ \forall\, i = 1, \ldots, M, \ \forall\, t = 1, \ldots, N \tag{13}$$

where N is the number of training data, and C_1 and C_2 are constants. Eq. (13) can be solved using Lagrange multipliers following the method of [1].

Analysis The voter collaboration is characterized by Eq. (12), which forces all voters to work together for classification. Effectiveness of each voter is governed by Eq. (11). It guarantees that each voter is learned from its corresponding homogeneous data. Eqs. (11) and (12) can accomplish decent classification performance.

σ **Selection** We solve Eq. (13) using different σs for $s(\cdot)$ in Eq. (7). In the final stage, we pick the σ with the minimum energy for the objective function Eq. (13).

4. Experiments

We create a new weather dataset that contains 10K images for training and testing. The dataset and our executable that implements the above collaborative learning will be made publicly available. In this section, we first describe how this dataset was constructed, and then report the performance of individual weather cues in the two-class weather labeling problem. Finally, we show results of collaborative learning with homogeneous voters.

4.1. Weather Image Dataset

Our new weather dataset contains sunny and cloudy images obtained from three sources: Sun Dataset [20], Labelme Dataset [17] and Flickr. The minimum and maximum dimensions of the images are respectively 600 and 1500.

To avoid bias, the helpers to collect and label images are unaware of the purpose or methods used in our experiments. They work with their own common sense. They collected 14,000 outdoor images in which sunny and cloudy images are roughly 50% each. The helpers contribute the same number of images each.

Very similar images are rejected using the following scheme. We compute the color histogram distance for all image pairs, and manually reject those, which are identical or highly similar. As a result, 1,121 sunny images and 812 cloudy images are rejected. Next, we ask 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 them are discarded. After this round, 5,467 sunny images and 5,612 cloudy images survive. Finally, we ask the third helper to pick 5,000 sunny and 5,000 cloudy images in the final dataset. Consequently, the 10,000 images in our dataset are unambiguous.

4.2. Classification Results

With the weather dataset constructed, we perform training and classification. We adopt cross validation where in each round, 80% of the data are selected randomly as the training set, with the remaining 20% belong to the testing set. We execute 5 rounds and report the mean and variance of the classification accuracy. On two-class labeling, even random guess can reach 50% accuracy. We normalize accuracy for better expressing the results. It is expressed as $\max\{(a-0.5)/(1-0.5), 0\}$, where a is the accuracy obtained traditionally. This normalized accuracy ranges in $[0, 1]$ and random guess generally gets zero.

4.2.1 Individual Cues and Scores

We use SVM to evaluate each individual weather cue used in Eq. (1). Table 1 tabulates the classification results. Intriguingly, Table 1 indicates that sky is the most important weather cue among the five. We believe 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, 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 in comparison to sky and contrast cue. We believe it is useful and the relatively low classification rate is mainly due to the fact that many images in our dataset simply do not exhibit detectable hazy phenomenon. 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% (normalized accuracy) when 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 thus exhibiting low dark-channel intensities.

Next, we evaluate individual existence scores, which are used to form Eq. (2). For each weather cue, we select s percent images with the highest existence score in the dataset and apply SVM classification on this image subset. Figure 10 shows the performance curves with varying s of each individual classifier. The plot indicates that our existence score design is effective – a weather cue gets more useful with a higher existence score.

4.2.2 The Proposed Learning Framework

We report our overall classification results and compare them with several methods.

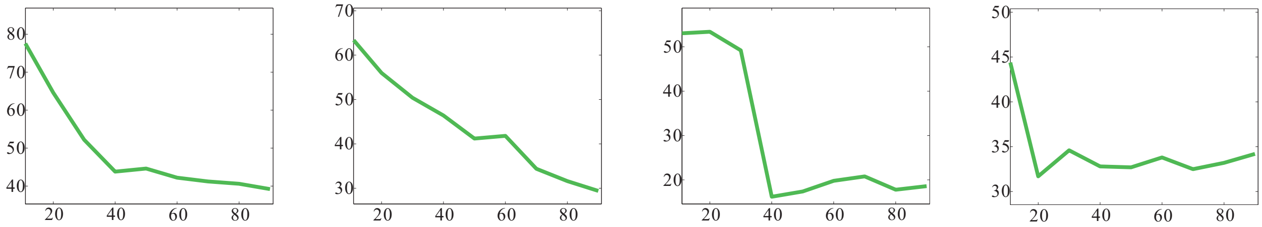
Comparison with Baseline Systems The first baseline is to implement SVM directly on the 621-D weather feature. We test both the linear and non-linear versions with different kernels and report the results with the best performance. The second baseline is the traditional Adaboost, which combines several classifiers to build a stronger classifier. We take each feature bin as a weak classifier. Another two baseline methods based on dictionary learning [14] are typical image classification methods, namely LLC [19] and ScSPM [22]. Table 2 tabulates the classification results. Figure 11 shows a few examples, where we test 5 different

	Sky	Shadow	Reflection	Contrast	Haze
normalized accuracy	39.3 ± 2.1	28.2 ± 2.4	23.0 ± 2.6	35.5 ± 2.2	30.2 ± 1.7

Table 1. Classification results (mean ± variance) using individual weather cues.

	SVM	Adaboost	LLC [19]	ScSPM [22]	Ours
Normalized accuracy	41.2 ± 2.2	36.4 ± 2.3	0.3 ± 0.1	0.2 ± 0.1	**53.1** ± 2.2

Table 2. Classification statistics (mean ± variance) of different classification methods.



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

Figure 10. (a)–(d) are respectively the performance curves of sky, shadow, reflection, and haze classifiers. The x-axis is the respective percentage of selected images set (with highest existence score) in the dataset. The y-axis is the respective labeling accuracy (in %).

σ values in Eq. (13), i.e., $\{0.5, 0.1, 0.01, 0.05, 0.001\}$, and select the best result with the lowest energy in Eq. (13).

Comparison with Related Methods We resort to other weather-related methods for comparison. The first related work is of Lalonde *et al.* [9]. Note that system is not designed for weather classification. Just one of the components, i.e., sun visibility prediction, can be regarded as a coarse weather estimator. We implement this component and test it on our dataset. Another two vehicle-based weather classifiers [21, 16] were also compared. Table 3 tabulates the classification statistics.

More Analysis For traditional image classification methods LLC [19] and ScSPM [22], 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 (c.f. Tables 1 and 2). We also find that the use of kernel SVM yields similar performance.

For the method of [9], the assumption that an outdoor scene is composed of ground, sky, and vertical surfaces may not be satisfied (see a few "exceptions" in Figure 11). Further, there is no consideration of the inhomogeneity of weather cues across images. For the work of [21, 16], the weather estimators are specially designed for driver assistance. They rely on vehicle-mounted image priors, which cannot properly deal with general natural images.

5. 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 at all given the great variety of outdoor images. The feature cues we used resonate well with our own common sense in judging weather conditions. Because not all of the feature cues are always available in images. The key to our computational framework is a collaborative learning strategy where only voters closer to the testing image information/structure are given more weight in classification. Our experimental results proved that this is a fairly effective and inspiring strategy that could be used in many computer vision tasks.

Our work is the first significant attempt in tapping into weather labeling given single images. While it shows promises, the problem is far from solved. Our framework is scalable to include more useful weather cues, and we expect better performance in the future with advances in shadow and haze detection. Two failure cases are shown in Figure 12. For (a), we check the extracted low-level features and conclude that our learning method is negatively impacted by the detected shadow and contrast cues. Our current approach is limited in labeling two weather types. More research needs to be engaged in generalizing it to labeling more weather types. We hope that this paper will spark interest and subsequent work along this line of research. Executable and the weather dataset are available at the project website.

References

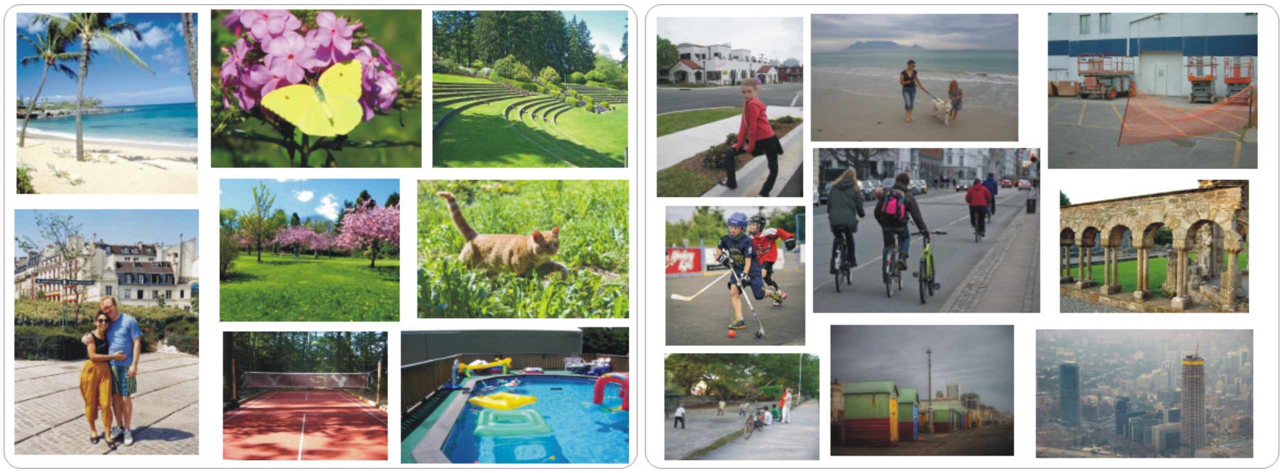
[1] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 1995.

[2] 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 *CVPR*, 2012.

[3] M. Fionn. A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, 1983.

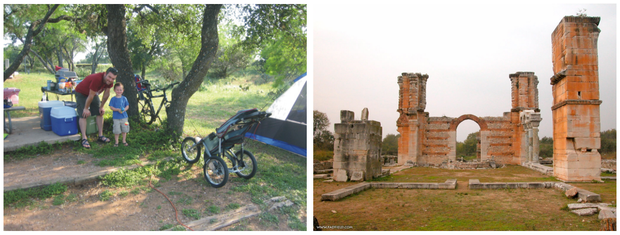
	Lalonde *et al.* [9]	Yan *et al.* [21]	Roser and Moosmann [16]	Ours
normalized accuracy	39.5 ± 2.3	24.6 ± 2.6	26.2 ± 2.3	**53.1** ± 2.2

Table 3. Classification statistics (mean ± variance) of different methods.



(a) (b)

Figure 11. Detection results. (a) Sunny images. (b) Cloudy images.



(a) (b)

Figure 12. Two failure cases. (a) Sunny image mis-detected as cloudy, and (b) cloudy image mis-detected as sunny.

[4] A. Geiger, M. Lauer, and R. Urtasun. A generative model for 3d urban scene understanding from movable platforms. In *CVPR*, 2011.

[5] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *CVPR*, 2009.

[6] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. In *CVPR*, 2009.

[7] 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 *IROS*, 2003.

[8] J.-F. Lalonde, A. Efros, and S. Narasimhan. Detecting ground shadows in outdoor consumer photographs. In *ECCV*, 2010.

[9] J.-F. Lalonde, A. Efros, and S. Narasimhan. Estimating the natural illumiation conditions from a single outdoor image. *IJCV*, 2012.

[10] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. In *CVPR*, 2006.

[11] D. Lin, C. Lu, R. Liao, and J. Jia. Learning important spatial pooling regions for scene classification. In *CVPR*, 2014.

[12] R. Liu, Z. Lin, F. De la Torre, and Z. Su. Fixed-rank representation for unsupervised visual learning. In *CVPR*, 2012.

[13] C. Lu, J. Shi, and J. Jia. Online robust dictionary learning. In *CVPR*, 2013.

[14] C. Lu, J. Shi, and J. Jia. Scale adaptive dictionary learning. *IEEE Transactions on Image Processing (TIP)*, 2013.

[15] S. Parizi, J. Oberlin, and P. Felzenszwalb. Reconfigurable models for scene recognition. In *CVPR*, 2012.

[16] M. Roser and F. Moosmann. Classification of weather situations on single color images. In *IEEE Intelligent Vehicles Symposium*, 2008.

[17] B. Russell, A. Torralba, K. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *IJCV*, 2008.

[18] L. Tao, L. Yuan, and J. Sun. Skyfinder: Attribute-based sky image search. In *SIGGRAPH*, 2009.

[19] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[20] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[21] X. Yan, Y. Luo, and X. Zheng. Weather recognition based on images captured by vision system in vehicle. In *CVPR*, 2009.

[22] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[23] P. Zhao, T. Fang, J. Xiao, H. Zhang, Q. Zhao, and L. Quan. Rectilinear parsing of architecture in urban environment. In *CVPR*, 2010.