J. Vis. Commun. Image R. 46 (2017) 233–249

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Camera as weather sensor: Estimating weather information from single images ☆

Wei-Ta Chu *, Xiang-You Zheng, Ding-Shiuan Ding

National Chung Cheng University, Chiayi, Taiwan

ARTICLE INFO

Article history:
Received 11 July 2016
Revised 6 December 2016
Accepted 14 April 2017
Available online 23 April 2017

Keywords:
Cross-platform data association
Weather property estimation
Weather modeling
Random forests
Landmark classification

ABSTRACT

We estimate weather information from single images, as an important clue to unveil real-world characteristics available in the cyberspace, and as a complementary feature to facilitate computer vision applications. Based on an image collection with geotags, we crawl the associated weather and elevation properties from the web. With this large-scale and rich image dataset, various correlations between weather properties and metadata are observed, and are used to construct computational models based on random forests to estimate weather information for any given image. We describe interesting statistics linking weather properties with human behaviors, and show that image's weather information can potentially benefit computer vision tasks such as landmark classification. Overall, this work proposes a large-scale image dataset with rich weather properties, and provides comprehensive studies on using cameras as weather sensors.

1. Introduction

Estimating image[1] properties from visual content is a fundamental step of various computer vision studies. For example, estimating image scene labels [1,2] facilitates image browsing and retrieval, and recognizing whether images were captured indoors or outdoors [3] facilitates place recognition. Recently, estimating geographic information from images [4] attracts much attention because various potential applications can be expected. In this paper, we advocate an image property that affects visual appearance of images and is well perceived by human beings, but has attracted little research attention for a long time: *weather information*.

By analyzing geographical or weather information of user-generated images, we could *unveil characteristics in the real world from images available in the cyberspace*. Comparing with geographical information, weather condition keep changing even at the same place, and thus we think weather variations across time periods provide richer information and give impact to wider fields. For example, by estimating weather information from images uploaded by users, the population's cameras can be viewed as weather sensors, and fine-grained weather monitoring can be achieved. Coupling estimated weather information with time/geographical information, explicit or implicit human behaviors can be discovered. For example, more people travel (and thus more photos taken) on weekends if it is sunny, and some places are especially attractive if the temperature is under −5 °C. Weather information can also serve as an important prior for many computer vision applications, e.g., object detection/recognition and scene categorization. Examples in Fig. 1 show that the Eiffel Tower has drastically different visual appearances in different weather conditions, which draws significant challenges on object/landmark recognition. Once weather properties can be estimated, an object detector/recognizer can adapt its parameters for different weathers, so that influence of visual variations can be reduced. In [5], better understanding weather properties facilitates robust robotic vision. Recently, modeling adaptive to weather conditions (or limited to lighting conditions) has been studied in lane detection and vehicle detection [6], and flying target detection [7]. The work in [8] also mentions that weather context may give clues to modeling appearance of objects.

One might think that weather forecast is a widely available service and there is no need to estimate weather properties from images. However, these general-purpose services do not enable deeper investigation like how the behaviors or tendency of (a bunch of) users relate to weather properties. Currently many posts on social media platforms include photos, and we think estimating weather properties from user's photos implicitly but intensely links social behaviors with weather. We may also expect that estimating weather properties from photos is trivial when the

☆ This paper has been recommended for acceptance by M.T. Sun.

* Corresponding author.

E-mail address: wtchu@ccu.edu.tw (W.-T. Chu).

[1] In this paper, we use *image* and *photo* interchangeably.

http://dx.doi.org/10.1016/j.jvcir.2017.04.002

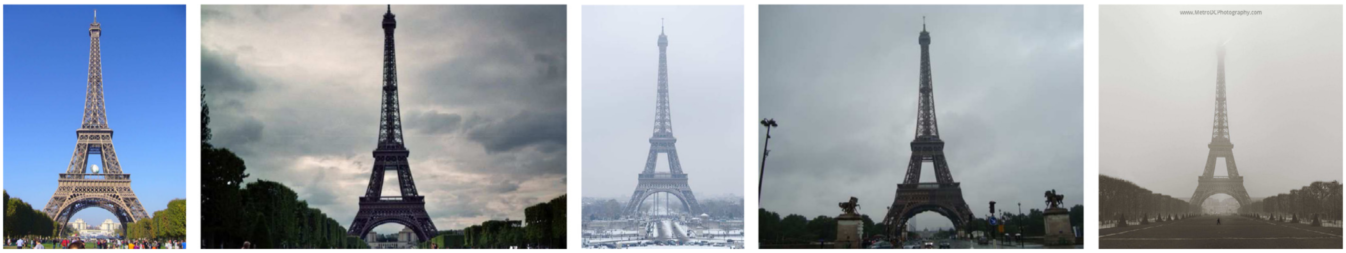


Fig. 1. The Eiffel Tower in different weather conditions. Left to right: sunny, cloudy, snowy, rainy, and foggy.

corresponding timestamps and geotags are available. However, according to the report provided by [9], many mainstream social platforms like Flickr and Facebook strip off uploaded photos' metadata, not to mention that geotags are often inaccurate [10].

Our goal is to construct computational models to estimate weather properties from single images. With rapid dissemination on social media platforms, photos with spatial (geotags) and temporal (timestamps) information can collectively reveal weather information around the world. In addition to ever-increasing computational power, new types of sensors and crowdsourcing techniques (in this case, photos captured by users) evolve weather forecasting [11]. In this work we particularly focus on: (1) How to collect a large-scale image collection associated with weather information and other useful metadata? (2) What explicit/implicit knowledge is embedded by such cross-platform image data? (3) How to construct computational models to estimate weather properties for any given image? (4) What kind of applications can be benefited by the estimated weather properties?

Contributions of this paper are summarized as follows.

- We collect a large-scale image collection associated with rich weather properties and with high visual dynamics. We show interesting statistics and human behavior bias based on the dataset.
- We adopt random forest classifiers/regressors to estimate weather properties, with the consideration of overfitting and ambiguity problems. We also compare performance obtained by hand-crafted features with that obtained by state-of-the-art deep features.
- We propose a realistic application, i.e., weather-aware landmark classification, that adopts weather information in the loop of classification. Few realistic applications have been proposed and verified in previous weather-related literature.

The rest of this paper is organized as follows. We review related literature in Section 2. In Section 3, we describe how to crawl weather information from a web-based weather platform, based on an existing large-scale image collection. A large-scale image dataset, named Image2Weather, collected across platforms then serves as the foundation of related researches. In Section 4, we show interesting statistics and correlations between metadata/visual features and weather information. Based on a variety of observed correlations, in Section 5 we propose and extract weather features and construct a classifier based on the random forest approach to recognize weather types. In addition, temperature estimation and humidity estimation are respectively formulated as regression problems, and are solved by the random forest regression approach. Experimental results and performance comparison with existing approaches are provided in Section 6. We develop a weather-aware landmark classification system to verify the benefit of weather information in Section 7, followed by the research summary in Section 8.

2. Related works

Recently estimating weather property from visual content has been envisioned to benefit computer vision applications. Narasimhan and Nayar [12] proposed one of the earliest works on studying visual manifestations of different weather conditions. Chromatic effects were modeled for images with fog or haze. They further presented the WILD database consisting of registered and calibrated images of a fixed outdoor scene to facilitate related studies [13]. To enhance driver assistance systems on vehicles, Roser and Mossmann [14] constructed an SVM classifier based on contrast, intensity, sharpness, and color features to classify images captured by the camera mounted on vehicles into clear, light rain, and heavy rain weather conditions. Shen and Tan [15] exploited the photometric stereo algorithm to recover illumination conditions of internet images coming from a specific scene. With illumination information indicating directional light and uniform light, weather condition of an image in the scene could be determined as sunny or cloudy. Chen et al. [16] classified weather conditions of panorama images captured by static cameras into sunny, cloudy, and overcast. They proposed a multiple kernel learning approach to find optimal linear combination of image features, and an active learning strategy was adopted to boost recognition performance.

For years Jacobs and his colleagues have a series of studies on scene attributes based on an image dataset called the Archive of Many Outdoor Scene (AMOS) [17], in which images were captured by static webcams over a long period of time. In [17], they investigated daily scene variations through analysis based on singular value decomposition. They also discovered that weather conditions, human activity, and change of season incur scene variations at longer timescales. In [18], they augmented the AMOS dataset with automatic scene alignment and object labeling. Based on the augmented dataset, they proposed that webcams installed across the earth can be viewed as image sensors enabling us to understand weather patterns and variations. Given a sequence of images, principal component analysis and canonical correlation analysis were sequentially used to predict wind velocity and vapor pressure. The AMOS+C dataset [19] was proposed as the first large-scale image dataset associated with weather information, which was extended based on AMOS by collecting weather data through timestamps and geo-location of images. With the AMOS+C dataset, Jacobs and his colleagues explored the relationships between image appearance, sun position, and weather conditions [19].

Most recently, Laffont et al. [20] constructed regressors to estimate scene attributes, including lighting, weather, seasons and subjective impressions for images captured by webcams. Crowdsourcing techniques were used to label attributes for images selected from the AMOS and Webcam Clipart [21] datasets. In [22], five weather features, i.e., sky, shadow, reflection, contrast, and haze, were extracted, and a collaborative learning framework was proposed to classify images into sunny or cloudy. A weather image dataset of moderate size (10 K images) was collected from

Flickr and the SUN dataset [23] for evaluation. Zhang et al. [24] proposed the most recent multi-class weather classification framework, where multiple features were extracted from images, and a multiple kernel learning approach was proposed to select the optimal feature set for classification. A dataset called MWI (Multi-class Weather Image) consisting of 20 K images was collected from web albums and films. Before this work, the authors also adopted camera as a air quality sensor [25]. Glasner et al. [26] found there is moderate correlation between pixel intensity/camera motion and temperature. By simply taking pixel intensities as features, temperature can be predicted by a regression model. They also showed simple intensity features outperform convolutional neural network features. Following [26], Volokitin et al. [27] argued that deep features should be more promising, and showed prediction accuracy can be improved 10% if appropriate fine tuning is conducted.

Comparing with existing works, our work has three distinct characteristics. First, we collect a large-scale image dataset shared by users around the world, which consists of over 180,000 photos captured from arbitrary viewpoints in unlimited amounts of lighting conditions. The AMOS dataset [17] is also a large-scale image collection, but photos in it were captured by fixed outdoor cameras. Second, much richer weather information was collected from online weather resources, and we estimate five weather types, temperature, and humidity from single images, comparing with estimating fewer weather types in existing works like [14,22]. Third, we investigate relationships between visual features and weather properties, as the basis of using visual features to construct weather property estimators. Interesting human behaviors can also be discovered through several statistics showing photo taking characteristics.

3. Building the Image2Weather dataset

3.1. Cross-platform data association

We need a large-scale image collection associated with heterogenous metadata to support rich image-weather association studies. In this work, we collect weather properties for each image from the Weather Underground website. Considering the most common potential applications, we mainly target the following weather properties:

- Five weather types: sunny, cloudy,[2] snowy, rainy, and foggy.
- Temperature: in terms of centigrade, generally from −25 °C to 45 °C.
- Humidity: from 0% to 100%.

According to our knowledge and availability of data, information related to the aforementioned weather properties includes, but not limited to:

- Taken time and taken location: Taken time indicates what season and what time on a day an image was taken, which is a factor highly correlated to an image's weather properties because weather conditions often periodically change. Taken location is also very important because weather is obviously related to a neighborhood.
- Textual annotation related to an image, such as tags, description, and image title: Text in these fields may implicitly or explicitly indicate weather properties, such as "hot", "cold", and "ski".
- Elevation: Temperatures at places with the same latitude would be significantly different because of higher elevation generally yields lower temperature.

To quickly build a convincing dataset, we crawl weather-related information based on an existing large-scale image collection, i.e., the European City 1 Million (EC1M) dataset [28], which has been widely used in image clustering and retrieval. Based on the URL and photo ID available in EC1M, we adopt the Flickr API to obtain the image itself, and its associated metadata such as taken time, taken location, and tags. Based on taken location (in the representation of latitude and longitude), the corresponding elevation information (in the representation of meters) is acquired through the Google Maps API. Fig. 2 shows the framework of our web crawler.

Based on longitude and latitude of an image, we utilize the Weather Underground API to retrieve more than thirty weather properties, while Table 1 shows a subset of these properties. For an image captured at time *t*, we find the temporally closest weather record captured by the spatially closest meteorological station. If the temporal distance between the weather record and the taken time is less than two hours, the retrieved weather properties are used to ”label” this image. Main properties to be estimated in this paper are weather types, temperature, and humidity, while other properties are left for future study.

Weather information on the Weather Underground website is from 60,000+ weather stations. With innovative forecast models and cross verification, it provides unrivaled amount of local neighborhood weather data. Fig. 3 shows the cumulative distribution of distances from our collected images to their closest meteorological stations. For almost all images, the distance from them to their closest stations is less than four kilometers, which means that the retrieved weather information, if available, is trustworthy.

The advantages of using the EC1M dataset as the base to collect cross-platform data association is worthy describing as follows. First, based on available photo IDs and URLs, we are able to quickly build a large-scale dataset associated with heterogeneous metadata. Second, photos in the EC1M dataset are mainly from big European cities, where meteorological stations are densely set up so that weather records are relatively richer and more accurate. Third, because the EC1M dataset was originally designed for landmark retrieval, with landmark information and the retrieved weather properties, researchers may be able to discover implicit correlation between weather and landmarks.

3.2. Data filtering

The EC1M dataset consists of 1,037,574 geo-tagged photos captured in 22 European cities. By excluding images with broken links and without corresponding weather properties, we totally collect 669,113 images from Flickr. For weather estimation, we focus on photos captured outdoors and where the sky region occupies more than ten percent of the whole photo. To filter out photos captured indoors, we extract CNN (convolutional neural network) features based on the MatConvNet [29] toolbox as image representation, and construct a support vector machine (SVM) classifier to achieve indoor-outdoor classification. We collected totally 23,900 indoor photos and 17,906 outdoor photos from [30], the SUN database [23], and Flickr for classifier construction and evaluation. The pre-trained model in MatConvNet has five convolutional layers and three fully-connected layers. In this work, we take output of the seventh layer as 4096-dimensional CNN features. Ninety percent of indoor and outdoor photos were randomly selected as the training data, and the remaining are for testing. According to our experiments, this classifier achieves more than 98% accuracy and enables us to largely eliminate indoor photos.

[2] Note that we can collect a variety of cloudy conditions from the website. We view "most cloudy" as cloudy, and view "partially cloudy" as sunny.

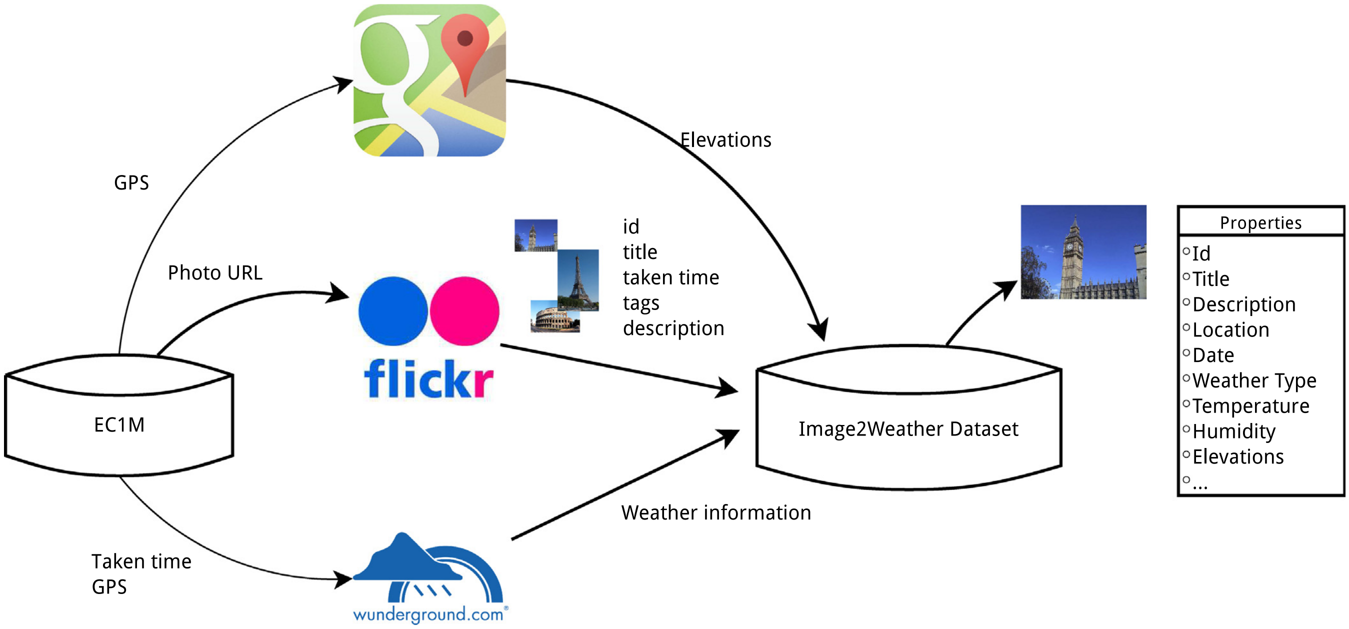


Fig. 2. The framework of our web crawler associating images with heterogeneous metadata (weather, tags, elevation).

Table 1
Weather properties obtained from the Weather Underground web site.

Property	Meaning
type	Weather types: clear (sunny), cloudy, snowy, rainy, foggy
hum	Humidity
date	Local time of the weather record
utcdate	Coordinated universal time
tempm	Temperature in terms of Centigrade
tempi	Temperature in terms of Fahrenheit
dewptm	Dew point temperature in terms of Centigrade
dewpti	Dew point temperature in terms of Fahrenheit
wspdm	Wind speed kph
wspdi	Wind speed mph
wgustm	Wind gust kph
wgusti	Wind gust mph
wdird	Wind direction in degrees
wdire	Wind direction description
vism	Visibility in km
wismi	Visibility in miles
pressurem	Pressure in mBar
pressurei	Pressure in inHg
windchillm	Wind chill in terms of Centigrade
windchilli	Wind chill in terms of Fahrenheit
heatindexm	Heat index in terms of Centigrade
heatindexi	Heat index in terms of Fahrenheit
precipm	Precipitation in mm
precipi	Precipitation in inches

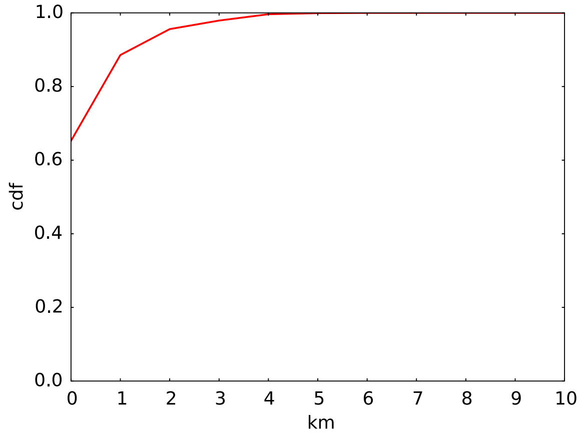


Fig. 3. Cumulative distribution function of distance between images and nearest weather stations.

Table 2
Statistics of the Image2Weather dataset after data filtering, and the number of photos in five different weathers in the final dataset.

D_1 = {Photos in EC1M}	1,037,574
$D_2 = D_1 \bigcap$ {Photos without broken links}	800,371
$D_3 = D_2 \bigcap$ {Photos with weather info.}	669,113
$D_4 = D_3 \bigcap$ {Outdoor photos}	293,071
$D^* = D_4 \bigcap$ {Photos with large sky}	183,798
Number of sunny photos	122,071
Number of cloudy photos	48,259
Number of snowy photos	1484
Number of rainy photos	10,428
Number of foggy photos	1556
Total	183,798

To detect sky regions in photos, we adopt the method proposed in [22] and classify each pixel into sky or non-sky. In addition to being used in data filtering, sky is also the region where several important visual features are extracted (described in Section 5).

Table 2 shows statistics of the Image2Weather dataset after different stages of data filtering (top half) and numbers of photos in five different weather types (bottom half). The final dataset (D^*) consists of 183,798 images in total. Overall, geographically the collected images span from 9.25° west longitude to 30.4° east longitude, and from 35° north latitude to 62.21° north latitude (covering most part of the Europe). The range of temperature is from $-25\,^{\circ}\mathrm{C}$ to $45\,^{\circ}\mathrm{C}$, and the range of humidity is from 4% to 100%. From this table we can clearly see that numbers of images in different weathers are imbalanced, which reflects that people tend to travel and take photos on fine-weather (sunny or cloudy) days. Table 3 shows images randomly sampled from the collected dataset, with five samples for each weather type. From these samples we can realize the challenge of weather type classification due to significant visual variations.

Table 3
Five random samples for each weather type. From top to bottom: sunny, cloudy, snowy, rainy, and foggy.

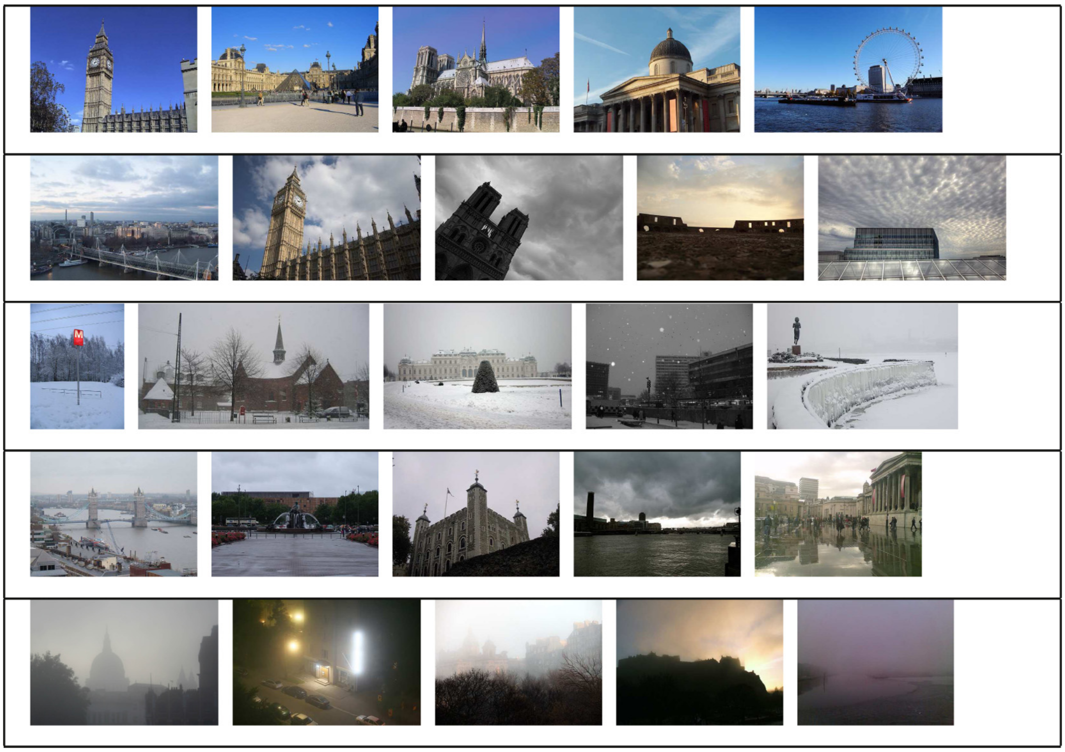


Fig. 4(a) and (c) respectively shows mean/standard deviation of temperature and humidity in different weather conditions. As we expect, the lowest temperature happens on snowy days, whereas the highest temperature happens on sunny days. Snowy, rainy, and foggy days have higher humidity. Fig. 4(b) shows a temperature distribution across different months. Like the distribution often seen in a travel guide book, higher temperature happens in summer. Fig. 4(d) shows that humidity is higher from October to March in the next year, which also matches with our impression on Europe's winter and spring. These distributions show that characteristics of the collected images are quite typical.

Table 4 compares our Image2Weather dataset with previous ones in terms of geographical information, weather information, and fixed/dynamic viewpoints. The Weather and Illumination Database (WILD) [13] contains images captured from static viewpoints. The Archive of Many Outdoor Scenes (AMOS) dataset [17] is a large-scale outdoor image dataset coming from static webcams but without clear weather metadata. The Archive of Many Outdoor Scenes with Additional Context (AMOS+C) dataset [19] consists of a subset of the AMOS dataset and associates weather data coming from a variety of sources. The dataset in [22] mainly contains images captured on sunny or cloudy days, and contains totally 10,000 images captured from dynamic viewpoints. The MWI dataset [24] consists of dynamic-view sunny, rainy, snowy, and haze images, and has around 20,000 images in total. Comparing with these datasets, our dataset consists of large-scale dynamic viewpoints images, richer weather information, and metadata collected from multiple platforms.

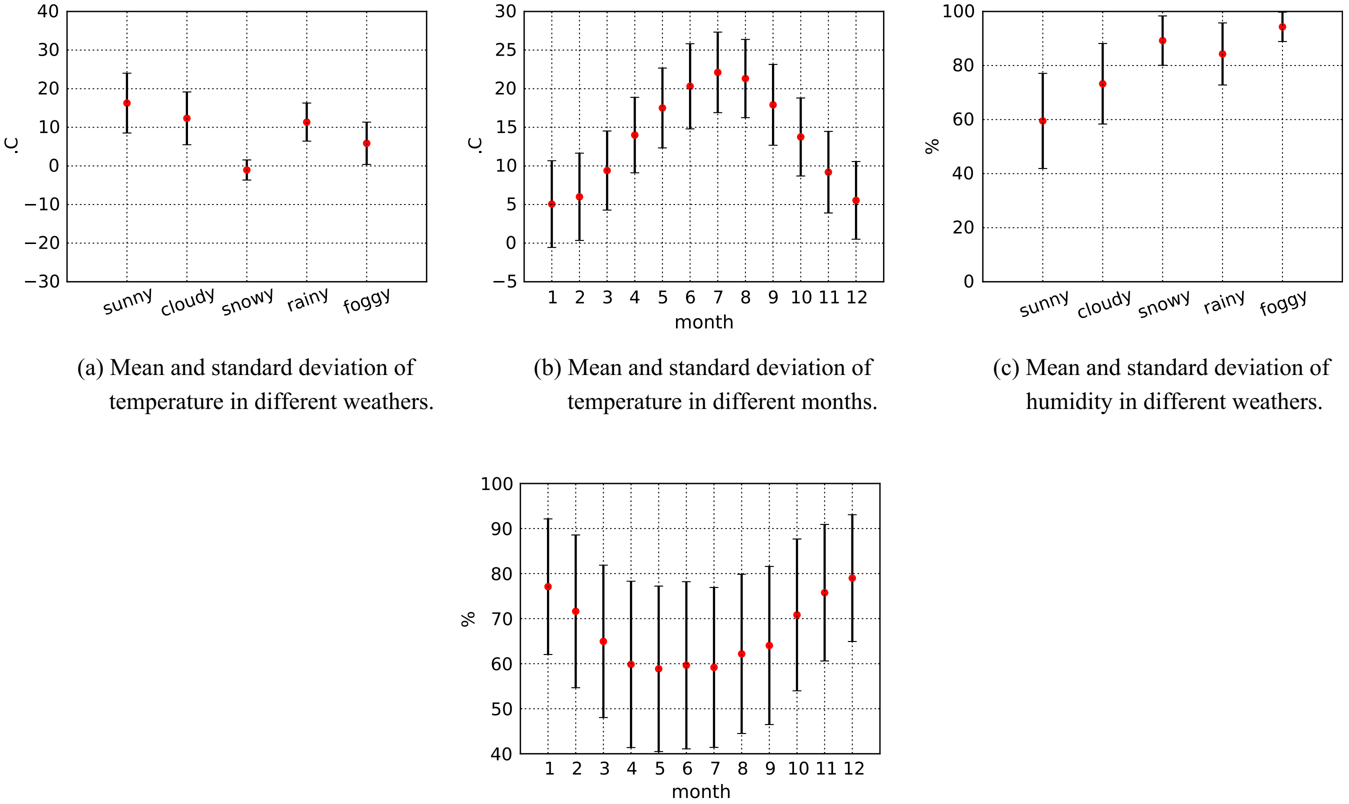
4. Data analysis

Based on the collected photos, we first discuss the relationship between weather types and photo taking behaviors. From the perspective of building estimation models, we then investigate how various features correlate with weather types, temperature, and humidity. Features used in this work include photo taken time, RGB color histogram, Gabor wavelet texture [31], intensity histogram, cloud features [32], local binary pattern (LBP) [33], contrast features, and haze features [22]. Detailed feature settings will be described in Section 5. Finally, we demonstrate correlation between weather properties through matrix visualization.

4.1. Relationship between weather type and photo taking behavior

The following statistics are obtained based on all images retrieved from Flickr and with weather information (669,113 images in the set D_3 mentioned in Table 2). Fig. 5(a) shows numbers of photos taken on different days. It is not surprising that photos taken on weekends are more than that taken on weekdays. Fig. 5(b) shows the ratio the number of images on Saturday to that on Monday, in different weather conditions. We can clearly see that, except for snowy days, the number of photos taken on Saturday is much larger than that on Monday. We conjecture that, even it is snowing, on Monday people still have to go out to work, while on snowy Saturday, people choose not to go out and take fewer photos. Fig. 5(c) shows the ratio of the number of photos taken outdoors to indoors. As we expect, more photos were taken on sunny days. Fig. 5(d) shows numbers of photos vs. temperature ranges. It can be seen that more photos were taken when the temperature ranges from 5 °C to 25 °C. Fig. 5(e) shows numbers of photos in different humidity ranges.

Table 5 shows the ratios of numbers of photos taken on sunny (cloudy) days to that taken on all days, and the ratios of numbers of views from sunny (cloudy) photos to the total number of views, at several famous landmarks. Most of them have similar patterns, i.e, there are more photos and views on sunny days. For the Eiffel Tower, the difference between sunny photos and cloudy photos is not as apparent as others. Interestingly, we can observe that photos of the Eiffel Tower captured on cloudy days attract more views.



(a) Mean and standard deviation of temperature in different weathers.

(b) Mean and standard deviation of temperature in different months.

(c) Mean and standard deviation of humidity in different weathers.

(d) Mean and standard deviation of humidity in different months.

Fig. 4. Statistics showing the relationships between temperature/humidity and weathers, and between temperature/humidity and time.

Table 4
Comparison of datasets.

Dataset	Geo.	Weather	Viewpoints	#images
WILD [13]	Yes	Yes	Static	3K
AMOS [17]	Some	No	Static	17M
AMOS+C [19]	Yes	Yes	Static	3.5K
[22]	No	Yes	Dynamic	10K
MWI [24]	No	Yes	Dynamic	20K
Ours	Yes	Yes	Dynamic	180K

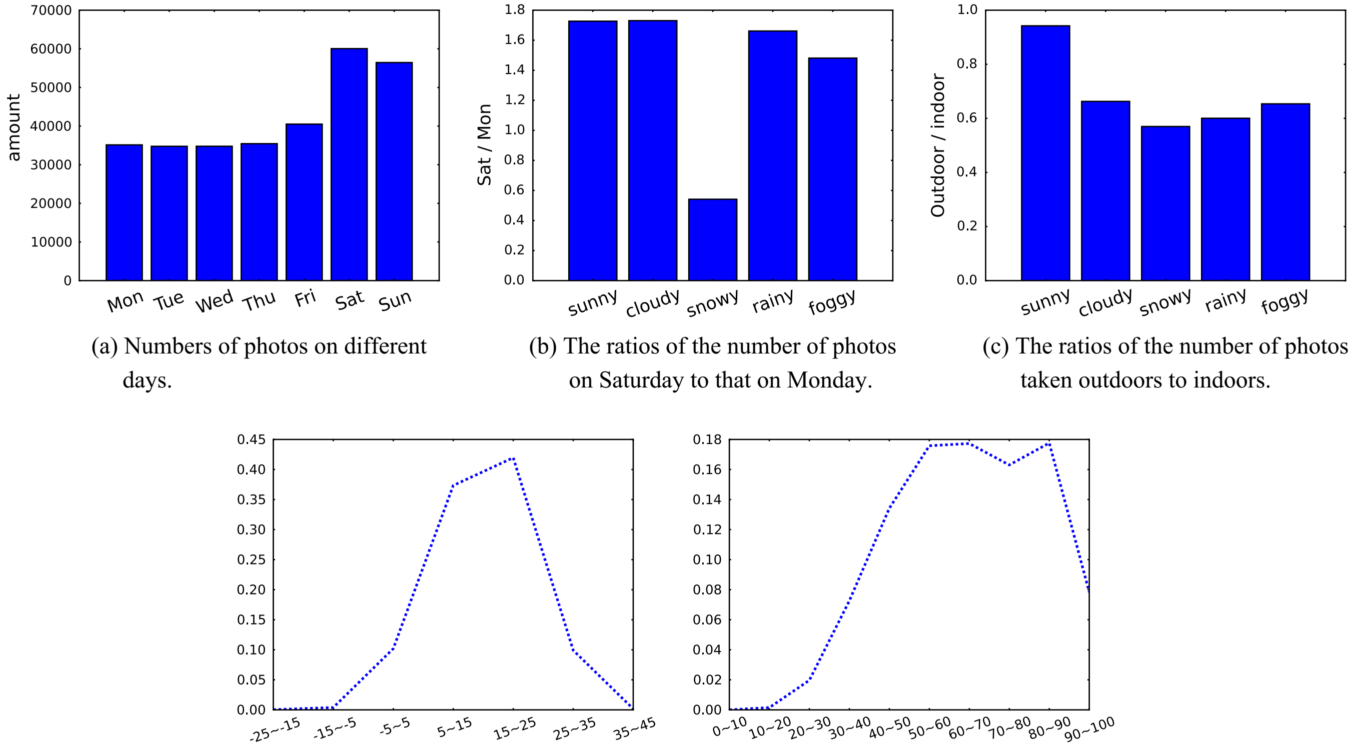
This discovery may inspire new research direction, e.g., some place is more popular in specific weather conditions.

4.2. Relationship between time distance and weather properties

Considering two photos that were taken on the same day, we can expect that weather properties of these two photos differ more if they were taken at larger temporal distance, e.g., one was taken in the early morning and another taken in the late afternoon. However, when we consider photos at a larger scale, periodicity of weather conditions may play an important role. For example, temperature of a photo taken at noon of someday in March may be similar to that of a photo taken at noon of someday in September, because temperature characteristics in spring and in fall are usually similar.

We especially focus on the month and the hour when a photo was taken, because month information (m) conveys which season this photo was taken, and hour information (h) is correlated with sunlight. Let us consider the time information of two photos as (m_1, h_1) and (m_2, h_2), respectively. To investigate the relationship between time distance and weather properties, in the meantime to consider periodicity of weather change, we collect average temperature of every month in European cities from the Weather Underground website. These data points are then fit by a polynomial curve f. The curve f acts as a function that transforms month information m into the value $\hat{m} = f(m)$, which indicates the estimated average temperature of the month m. Similarly, we can also fit average temperature of every hour and fit them with a curve (function) g, which then transforms hour information h into the value $\hat{h} = g(h)$. The time distance between the considered two photos is then calculated by $\sqrt{w_1(\hat{m}_1 - \hat{m}_2)^2 + w_2(\hat{h}_1 - \hat{h}_2)^2}$. The weight $w_1 = 10w_2$ is designed to emphasize the distance between months.

Fig. 6(a) shows probabilities of sharing weather types vs. time distances between photo pairs. The x-axis denotes the time distance between photo pairs, and the y-axis denotes the probability of sharing the same weather type. For example, the blue dot in this figure shows that the probability of two photos belonging to the same weather type is 0.5, if their time distance is 2. From this figure, we see that the probability of sharing the same wether type decreases as the time distance between photos increases. Fig. 6 (e) is a heat map showing the relationship between temperature distance and time distances. The x-axis denotes the time distances between photo pairs, and the y-axis denotes the temperature distances between photo pairs. For example, the white dot in this figure shows that, when the time distance between a photo pair is 1 (x-axis), the probability of their temperature distance ranges from 0 to 1 degree centigrade (y-axis) is around 0.27. The white square, on the other hand, shows that, when the time distance between a photo pair is 5 (x-axis), the probability of their temperature



(a) Numbers of photos on different days.

(b) The ratios of the number of photos on Saturday to that on Monday.

(c) The ratios of the number of photos taken outdoors to indoors.

(d) The ratios of the number of photos within different temperature ranges.

(e) The ratios of the number of photos within different humidity ranges.

Fig. 5. Numbers of photos on different days, different environments, in different weathers.

Table 5
Ratios of the number of sunny or cloudy photos to total number of photos; and ratios of the number of views from sunny or cloudy photos to total number of views.

Landmark	Ratio of photos	Ratio of views
Colosseum	Sunny: 0.79 Cloudy: 0.17	Sunny: 0.77 Cloudy: 0.16
Big Ben	Sunny: 0.64 Cloudy: 0.28	Sunny: 0.67 Cloudy: 0.25
Eiffel Tower	Sunny: 0.49 Cloudy: 0.44	Sunny: 0.44 Cloudy: 0.51
Notre Dame	Sunny: 0.50 Cloudy: 0.41	Sunny: 0.52 Cloudy: 0.41
London Eye	Sunny: 0.64 Cloudy: 0.27	Sunny: 0.67 Cloudy: 0.25

distance ranges from 0 to 1 degree centigrade (y-axis) is around 0.05. When the taken times of two photos are close, the probability of their temperature distance less than two degrees centigrade is much higher than other cases. Similarly, Fig. 6(i) is the heat map showing the probability of humidity distance within a range versus time distance between photos, and a trend similar to Fig. 6(e) can be seen. Particularly when the time distance between two photos is small, the probability of humidity distance less than 5% is much higher than other cases.

4.3. Relationships between visual feature distances and weather properties

Fig. 6(b) shows the probability of sharing weather type versus photo pairs' RGB color histogram distances (measured by Euclidean distance). We can see smaller distance between color distributions indicating higher probability of sharing weather type. However, by comparing Fig. 6(b) with Fig. 6(a), this effect is relatively moderate and noisy, showing that the color feature is not as robust as the time feature to estimate weather types. Fig. 6(f) is the heat map showing the relationship between color histogram distance (x-axis) and temperature distance (y-axis) within a range. We see that, if color histogram distance is less than 0.2, the probability of temperature distance less than one degree centigrade is much higher than other cases. This confirms that color distribution distance would still be an useful feature to estimate temperature. Fig. 6(j) also shows the characteristic on estimating humidity.

What are the relationships between other visual feature distances and weather properties? Generally, the trends derived from other visual features are similar to that from color distribution, while the strength would be different. Comparing the heat maps in Fig. 6(h) and (l) with Fig. 6(b) and (k), respectively, we see that intensity features would be more reliable than texture features to estimate temperature and humidity because the patterns are more concentrated.

The same analysis can be done for other features. We skip detailed illustration because of space limitation. The influences of different features on weather properties are different and complex. In Section 5, we will construct estimation models that automatically adopt features with different extents learned from training data.

4.4. Correlation between weather properties

With various weather properties listed in Table 1, each image can be viewed as a high-dimensional vector in the representation of tens of weather properties. To discover correlation between weather properties, we utilize Generalized Association Plots

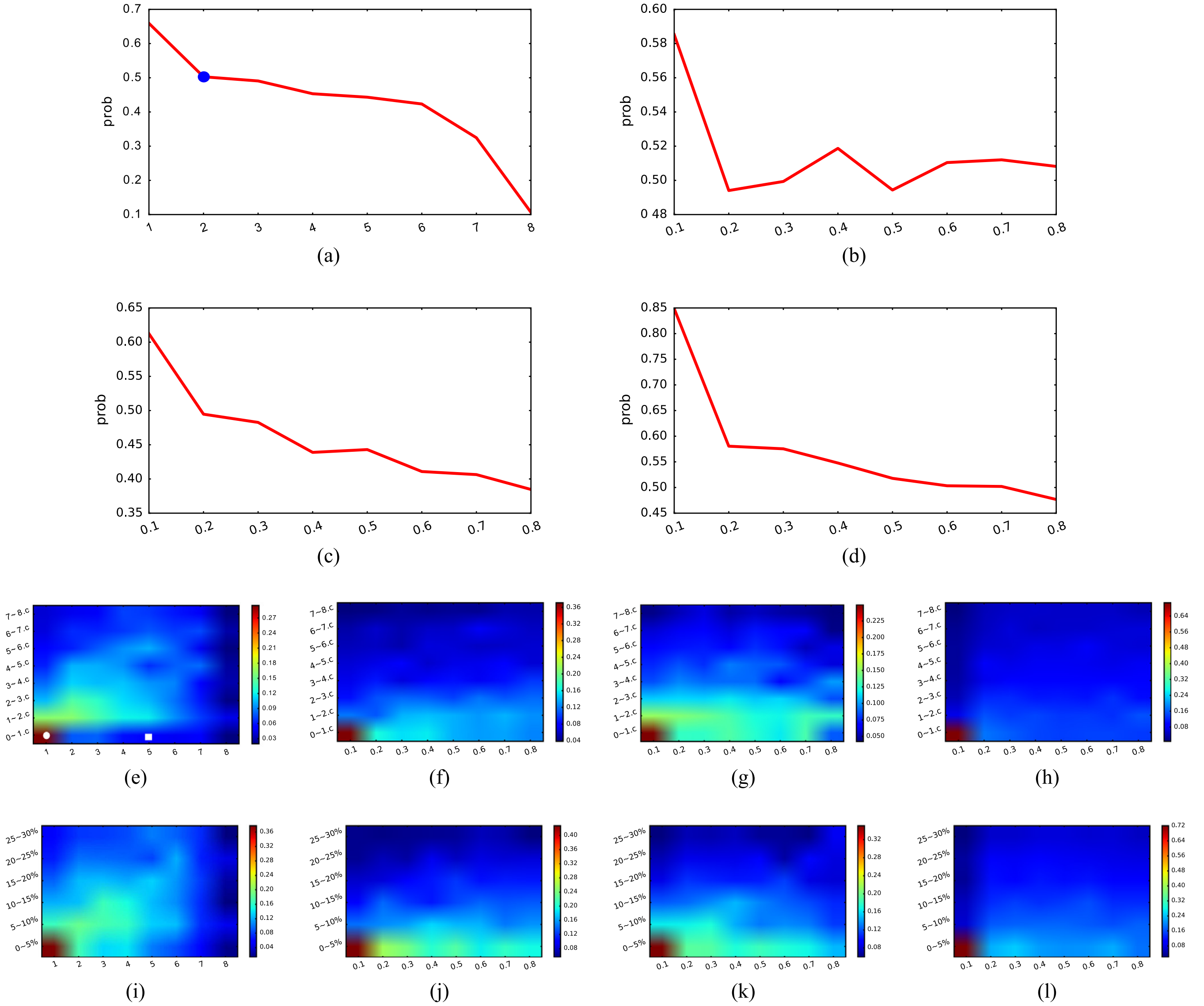


Fig. 6. Relationships between various properties and weather properties. (a)–(d) The prob. of sharing the same weather type vs. time/color/texture/intensity distances, respectively. (e)–(h) The prob. of temperature distance within a range vs. time/color/texture/intensity differences between photo pairs, respectively. (i)–(l) The prob. of humidity distance within a range vs. time/color/texture/intensity differences between photo pairs, respectively. These figures are better viewed in color version. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(GAP) [34] to visualize high-dimensional data. GAP is a exploratory data analysis tool for matrix visualization and clustering. It was designed to facilitate exploration of high-dimensional data with suitable color projection and clustering/seriation algorithms, without preset dimension reduction methods.

Processing in GAP mainly consists of four steps. Given an n by p data matrix Z, indicating that there are n images in the representation of p-dimensional vectors, the first step is to construct a p by p matrix C, where the (i, j)th entry indicates similarity or correlation between the ith and the jth properties, and an n by n matrix R, where the (s, t)th entry indicates similarity or correlation between the sth and the tth images. The second step is to appropriately permute entries in C and R so that similar/highly correlated entries are placed in a neighborhood. Entries in the data matrix are also permuted according to the permutations for C and R. Third, with appropriate clustering, entries in Z, C, and R are clustered into groups so that we can more clearly see relationships between weather properties. Finally, each group can be briefly represented as a value calculated based on median of mean of entry values in this group. Details of GAP please refer to [34].

In this work, we randomly select 200 images from each weather type, and each image is represented as a row vector constituted by 28 weather properties (23 properties and 5 weather types) shown in Table 1 plus time and elevation information, yielding a 1000 × 30 data matrix Z. With GAP, we measure Pearson correlation between entries to construct matrices C and R. Agglomerative hierarchical clustering with the average-linkage method for measuring distance between nodes is used to permute matrices. Fig. 7(a) shows the overall visualization of sorted and partitioned matrices Z, C, and R, while C is too small to be clearly seen. From the sorted R (relationship between images), we see that images can be clearly clustered into four groups (the four blocks along the diagonal from top left to right bottom in R), corresponding to snowy images, foggy images, rainy images, and mixture of sunny and cloudy images, respectively. In terms of weather properties, sunny and cloudy images are quite similar, and they are mixed at the right-bottom group (the fourth block along the diagonal) of *R*. It is often hard to define a photo with thin cloud as sunny or partially cloudy. In terms of weather properties like wind speed and air pressure, sunny photos and partially cloudy photos are quite similar. From the third block along the diagonal, we also found that some rainy photos and cloudy photos are quite similar.

Fig. 7(b) shows the enlarged matrix *C* demonstrating the relationship between weather properties. It can be clearly seen that wdird, rain, vism, visi, wspdm, wspdi, wgustm, and wgusti are positively correlated. We especially notice the close relationship between visibility and wind speed. Another very positively correlated group includes dewpti, dewptm, tempm, tempi, windchilli, and windchillm, which are all related to temperature. Regarding the relationships between weather types and other properties: sunny is positively correlated with temperature (tempm and tempi), and is negatively correlated with humidity; cloudy is positively correlated with visibility (vism and visi); snowy is positively correlated with elevation, and is negatively correlated with temperature (dewpti, dewptm, tempm, tempi, windchilli, and windchillm); rainy is positively correlated with wind (wgustm and wgusti) and temperature (windchilli and windchillm); foggy is positively correlated with humidity and air pressure (pressurem and pressurei), and is negatively correlated with visibility (vism and visi).

5. Weather property estimation

We construct computational models to estimate weather properties from single images, given mainly the image itself and when it was taken. Although there are many weather properties, in this paper we focus on weather type, temperature, and humidity estimation.

5.1. Features for weather property estimation

Eight types of features, including 2-D photo taken time (month and hour), 64-D RGB color histogram, 32-D Gabor wavelet texture [31], 64-D intensity histogram, 64-D cloud features [32], 64-D local binary pattern (LBP) [33], 84-D haze features [22], and 171-D contrast features [22] are used to describe a photo. Table 6 summarizes the used features with the corresponding type, extraction region, and dimensionality.

We extract time features, haze features, and contrast features from the whole photo, and extract other visual features from only the sky region. For time features, we especially focus on “month” and “hour” of the taken time of each photo. A photo’s time information is thus represented as a 2-D feature vector. For RGB color histograms, the RGB color space is equally divided into 64 ranges, and thus a 64-D feature vector (with normalization) is constructed. For Gabor texture features, image pixels’ intensity are transformed into the frequency domain, which is then decomposed into 16 ranges by the Gabor Wavelet functions with four scales and four orientations. Mean and standard deviation of the magnitude of the transform coefficients in each range are used to represent each frequency band, and are then concatenated to form a 32-D texture feature vector. Similar to color histogram, dynamics of intensity values are equally quantized into 64 ranges, and then a 64-D intensity histogram is obtained. For cloud features, the normalized blue/red ratio, saturation, and the Euclidean geometric distance [32] are extracted to represent each pixel. The three features of all pixels in the sky region constitute a feature space, which is equally divided into 64 ranges, and then a 64-D histogram is constructed to describe cloud characteristics. For LBP features, a local binary pattern is extracted to represent each pixel, and the space constituted by all LBPs extracted from the sky region is equally divided into 64 ranges to construct a 64-D LBP histogram. For haze features, dark channel prior [35] is first calculated for each pixel. A photo is partitioned based on the spatial pyramid scheme, i.e., uniformly partitioned into $2^2, 4^2$, and 8^2 non-overlapping regions to obtain 84 sub-regions. The median values of dark channel intensities in these sub-regions are concatenated as a 84-D haze feature vector [22]. For contrast features, we first calculate all saturation percentiles from all pixels’ saturation values. Let p_i denote the *i*th saturation percentile. All saturation percentile ratios can then be calculated as $\{r|r = p_i/p_j, \forall i > j\}$, where *i* and *j* are multiples of five. Totally 171 percentile ratios are obtained to form a 171-D contrast feature vector [22].

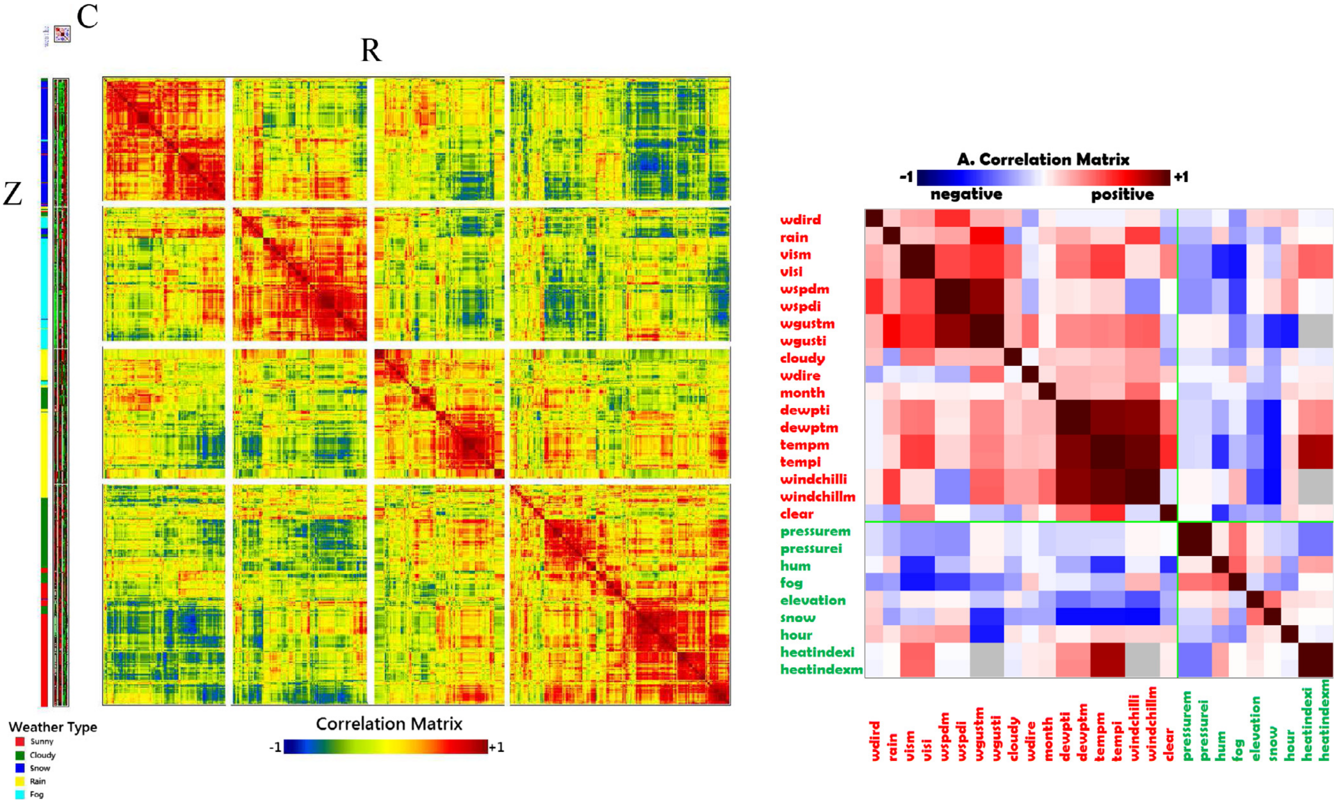
5.2. Weather type classification

5.2.1. Random forest for weather type classification

We construct a random forest classifier [36] to estimate weather type of a given image. A random forest classifier is composed of a number of decision trees, where each decision tree is a simple and weak classifier. By combining results of a large number of weak classifiers, a robust classification result can be obtained. To construct and evaluate this random forest classifier, for each weather type we randomly sample 1100 photos from the collected Image2Weather dataset. The random-split scheme is used for training and testing. That is, for each run 1000 photos are randomly selected from each weather type for training, and the remaining 100 photos are for testing. We conduct ten runs of training and testing, and report the average classification accuracy in the evaluation section.

Let $X = \{x_i\}$ denote a subset of the training feature vectors, and $x_{i,j}$ denote the *j*th feature value (*j*th dimension) of the *i*th feature vector. Each node of a decision tree is a binary classifier, which partitions training samples into two groups. For a partition, one of the feature dimensions is randomly selected. Based on the selected dimension *k*, we compare a feature vector x_i’s feature value $x_{i,k}$ with a threshold ϵ_k to determine whether x_i is categorized into the left subtree or the right subtree. Based on the dimension *k* and the threshold ϵ_k, all training samples in *X* are checked, and these samples are categorized into two groups, called Y_1 and Y_2. To evaluate homogeneity of each group, a measure called Gini impurity [37] is adopted: $Gini(Y_j) = 1 - \sum_{i=1}^{W} p(i|Y_j)^2$, where *W* is the number of weather types, i.e., $W = 5$ in this paper. The value $p(i|Y_j)$ is the probability of weather class *i* in the group Y_j. If the distribution of weather class in a partition is imbalanced, $p(i|Y_j)$ gets larger and the Gini impurity value gets smaller. Overall, the Gini impurity values of the left group Y_1 and the right group Y_2 obtained by the partition T_k are integrated by weighted summation: $Gini(T_k) = Gini(Y_1)|Y_1| + Gini(Y_2)|Y_2|$, where $|Y_j|$ means the number of feature vectors in the group Y_j, normalized by the total number of feature vectors in all groups. Smaller Gini impurity value means higher discriminative power of the partition T_k, which is the partition obtained based on the selected dimension *k* and the threshold ϵ_k. Fig. 8 shows examples of Gini impurity values of different partitions. By the first partition, images with different wether types are grouped together in both subtrees. By the second partition, sunny photos are grouped in the left subtree, and two of three photos in the right subtree are cloudy. The Gini impurity value of the second partition is much smaller than the left one.

If the number of feature dimension is *D*, at each node totally $\sqrt{D}$ dimensions are randomly selected and $\sqrt{D}$ random thresholds are used for partitions. That is, we totally try $\sqrt{D}$ different partitions at each node, and determine the best partition by finding the one with the smallest integrated Gini impurity value. The same process repeats at each child node to further (optimally) partition a group into two subgroups. This partition procedure keeps going until



(a) Overall visualization.

(b) The enlarged matrix C showing correlation between weather properties.

Fig. 7. Correlation analysis based on GAP (better seen in color version). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
The used features with the corresponding type, extraction region, and dimensionality.

Features	Type	Global/sky	Dimensionality
Time	Metadata	Global	2
RGB color histogram	Visual	Sky	64
Gabor wavelet	Visual	Sky	32
Intensity histogram	Visual	Sky	64
Cloud features	Visual	Sky	64
Local binary pattern	Visual	Sky	64
Haze features	Visual	Global	84
Contrast features	Visual	Global	171

only one feature vector is in a subtree, or in a subtree all data points belong to the same class. Finally, a single binary decision tree is obtained. In our work, totally 100 decision trees are constructed to constitute the random forest.

When a testing sample is given, the feature dimensions same as the training process are compared, and the test sample is classified by traversing a decision tree. One decision tree gives a classification result. The final classification result is determined by majority voting based on the results of all decision trees.

5.2.2. Improved random forest classifiers

Random forests have been widely employed to handle various classification and regression problems. However, two practical challenges often appear and harm performance: overfitting and ambiguity [38]. Because the dimension of the image feature space is typically high, the random forests constituted by an ensemble of decision trees often encounter the overfitting problem. That is, we may obtain pretty low error rate at the training stage, but much higher error rate is yielded when practical testing data are

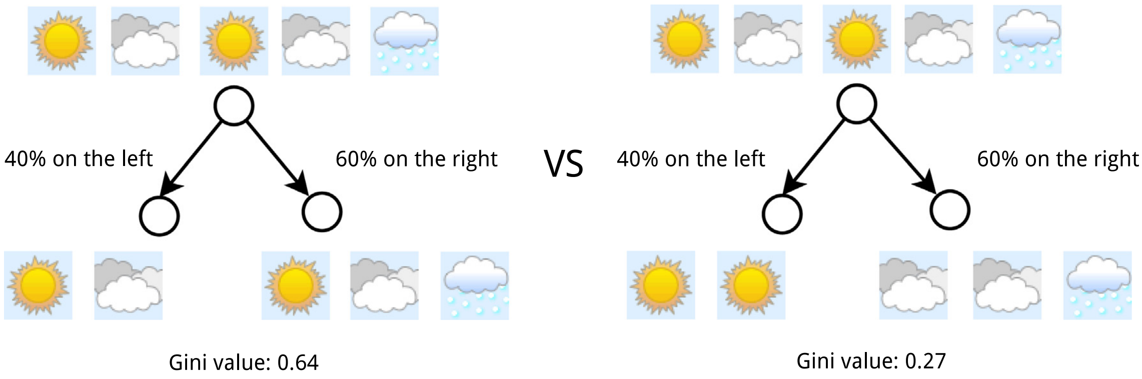


Fig. 8. Examples showing Gini impurity values of different partitions.

evaluated. About the ambiguity problem, because the output of a random forest is derived from averaging (or majority voting) results of individual trees, the distinctiveness between different trees are suppressed. One may want to emphasize the influence of "good" trees and prune the bad ones.

To mitigate the overfitting problem, we apply principal component analysis (PCA) and kernel PCA (KPCA) to reduce feature dimension. After dimension reduction, the search space of tree construction is shrunk, and thus higher generality can be obtained [39]. We will show the effectiveness of dimension reduction in the evaluation section.

To mitigate the ambiguity problem, we implement the trimmed bagging method proposed in [40]. Before averaging or voting results of decision trees, we first sort them by classification error rate in an ascending order. Only results of $\alpha\%$ of the trees with the smallest error rates are averaged. This simple idea discards a portion of "worst" trees to make final results more reliable. We will also show the effectiveness of trimmed bagging in the evaluation section.

5.3. Temperature and humidity estimation

We formulate temperature estimation as a regression problem, and use random forest regressors to estimate temperature. Given the training set, a random forest consisting of a number of decision trees is constructed. At each node of a decision tree, we measure how well the image set is partitioned into groups by mean squared error: $MSE = \frac{1}{M}\sum_{i=1}^{M}(\bar{t} - t_i)^2$, where M is number of photos located in the same group, $\bar{t}$ is the mean temperature of photos in the same group, and t_i is the temperature of the ith photo. At each node, the goal is to partition a given photo set into two groups such that the sum of MSEs of two groups are the smallest. This is achieved by randomizing thresholds to partition for 100 times and selecting the best one at each node.

Overall, we construct a random forest consisting of 100 decision trees. Given a photo, we traverse each decision tree from the root to the leaf, and calculate the mean temperature of training images at the leaf as the estimated temperature. The 100 estimated temperatures from 100 decision trees are averaged to be the final estimation value. Similarly, the humidity estimation task can also be formulated as a regression problem and solved by random forest regressors, while the MSE mentioned above is calculated based on humidity values.

5.4. Weather property estimation with the help of other weather properties

Three weather properties mentioned in Sections 5.2 and 5.3 are estimated separately based on weather features. However, weather type, temperature, and humidity are often correlated, e.g., foggy days have higher humidity, and sunny days often have higher temperature. Better weather type classification, for example, may be achieved if temperature and humidity can also be considered as features.

To verify the help of other weather properties, we propose a two-stage estimation framework. Given a test photo, we know nothing about the three weather properties in the beginning. We separately estimate weather type, temperature, and humidity as mentioned previously, and denote them as W_0, T_0, and H_0, respectively. To improve weather type classification, we concatenate weather features with T_0 and H_0, and re-train the random forest classifier for weather type classification, which then yields the improved classification result W_1. Similarly, by further considering W_0 and H_0, improved temperature estimation result T_1 can be obtained by re-training the model. Also, by further considering W_0 and T_0, improved humidity estimation result H_1 can be obtained. We will show performance comparison W_0 vs. W_1, T_0 vs. T_1, and H_0 vs. H_1, respectively, in the evaluation section.

6. Evaluation

6.1. Performance of sky detection

Because most features are extracted from the sky region, performance of sky detection influences classification results. In this work, we detect the sky region based on the method proposed in [22]. A given photo is divided into 15 × 15 patches, from each of which the SIFT descriptor and mean HSV colors are extracted. Based on features extracted from the training data (including positive and negative samples), a random forest classifier is constructed to classify each input patch into a sky patch or not. The detected sky patches are then viewed as seeds for the succeeding graph cut segmentation algorithm. After that, each image pixel can be classified into a sky pixel or not.

To verify sky detection performance, we randomly selected 180 photos from the Image2Weather dataset and manually defined their sky regions. Let $\mathscr{D}$ and $\mathscr{G}$ denote the set of detected sky pixels and the set of truth sky pixels, respectively, accuracy of sky detection is measured by $\frac{|\mathscr{D}\cap\mathscr{G}|}{\max(|\mathscr{D}|,|\mathscr{G}|)}$, where $|\cdot|$ denotes the number of pixels in a set. Based on the selected photos, we averagely obtain 76.6% accuracy in sky detection, which is robust enough to be the basis for feature extraction and the succeeding studies. Fig. 9 shows sample results of sky detection.

Note that we adopt the method proposed in [22] because of its simple implementation. More recent methods like the deep ensemble approach [41] can also be used to detect the sky region.

6.2. Performance of weather type classification

6.2.1. Effectiveness of different features

In Section 5.1, the two-dimensional time feature is a meta feature, while the others are all visual features. Among visual features,

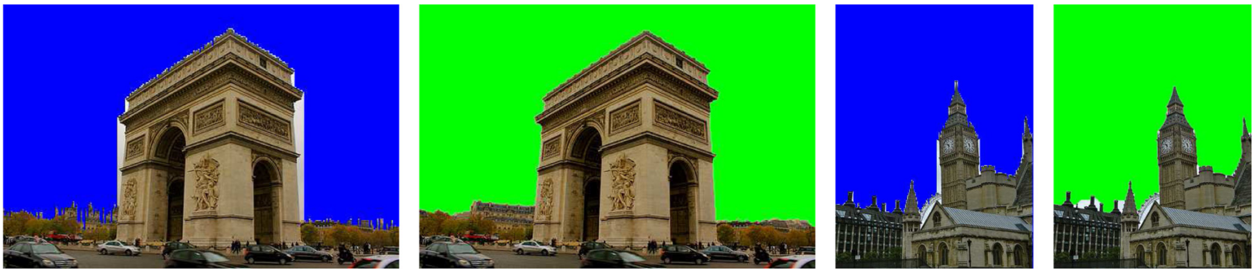


Fig. 9. Sample results of sky detection. The blue regions in the first and the third figures are detected sky regions, and the green regions in the second and the fourth figures are manually defined sky regions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 7
Average weather type classification accuracy obtained based on different weather-related features.

	Visual features (local)	Visual features (global)	Visual features (local + global)	Time feature	All features
Accuracy	0.55 ± 0.023	0.46 ± 0.022	0.56 ± 0.014	0.43 ± 0.013	0.58 ± 0.017

Table 8
Average numbers of times different features are used in tree partition.

Features	Average number
Time feature	529.0
RGB color histogram	25.8
Gabor wavelet	168.9
Intensity histogram	139.7
Cloud features	15.9
Local binary pattern	261.2
Haze feature	254.0
Contrast feature	162.7

haze features and contrast features are extracted globally from the whole photo, and the other visual features are locally extracted from the sky region.

Table 7 shows average accuracy of weather type classification based on different types of features. As can be seen from this table, jointly considering local and global visual features yields better performance. If visual features and the meta feature are jointly considered, the best performance can be obtained by a significant margin over visual features only.

A feature selection scheme is actually embedded in the construction procedure of random forests. At each node, several feature dimensions are randomly selected, and the partition causing the minimum Gini value is then determined. If some feature dimensions are more often selected to do partition, we would say these feature dimensions are more important in classification. Therefore, to show feature effectiveness we intentionally train 100 random forest classifiers based on the same training set, and check the average number of times each feature vector is selected to do partition at a node. Table 8 shows the average numbers presenting interesting characteristics. The most important feature is time, which provides important information on periodic change (month) and the condition of sunlight (day). The most important visual features are LBP and haze features. The former is a local feature, and the latter is a global feature. RGB color histogram plays a less important role, and surprisingly, the cloud features proposed in [32] is not reliable.

In addition to the weather-related features, we also notice that many current works show convolutional neural network (CNN) features largely surpasses hand-crafted features in many image classification works [42–44]. To verify the effect of CNN features on weather type classification, we utilize the MatConvNet package [29] with the pre-trained model obtained based on the ImageNet ILSVRC-2012 dataset to extract CNN features from the detected sky region or the whole image. There are five convolutional layers and three fully-connected layers in the CNN model. According to our experiments, we take output of the sixth layer as image description, which is 4096-dimensional.

Table 9 shows average classification accuracy based on different feature combinations. Random forest classifiers were separately trained based on the proposed weather features, CNN features extracted from the sky region, CNN features extracted from the whole image, and concatenations of two types of features. From this table we see that CNN features, no matter from the sky region or the whole image, do not outperform the weather-related features. This may attribute to (1) high intra-class variation and (2) relatively fewer texture information from the sky region (sky-only CNN features) and, on the contrary, too many noises from the whole image (whole-image CNN features). The proposed weather features, though not perfect, yield more robust classification results.

6.2.2. Effectiveness of dimension reduction

We verify that reducing dimension of feature vectors mitigate the overfitting problem of random forest classifiers. Fig. 10(a) shows the relationship between average weather type classification accuracy and the dimensions of reduced feature vectors. We see that, average accuracy can be clearly increased if the dimension of feature vectors is appropriately reduced (the original dimension is 564, Section 5.1), no matter by PCA or KPCA. Based on PCA, the best performance (0.594) can be obtained when the number of dimension is reduced to 50. Similarly, based on KPCA (rbf kernel), the best performance (0.596) can be obtained when the number of dimension is reduced to 50. In the following, we take the accuracy 0.596 as the baseline for further discussion.

6.2.3. Effectiveness of trimmed bagging

After dimension reduction, we further consider trimmed bagging to mitigate the ambiguity problem of random forest classifiers. We also conduct the experiments based on the ten-fold cross validation scheme. At each fold, 10% of the trees with worst performance are discarded, and the test accuracy is calculated. The average classification accuracy over ten folds are finally obtained. Overall, average accuracies of the random forest classifiers with and without trimmed bagging are 0.599 and 0.596, respectively. Although this improvement is very marginal, the p-value of this performance difference is 0.0375, showing satisfactory statistical significance. We expect that more advanced tree pruning methods can be adopted to improve performance further.

6.2.4. Results of the two-stage framework

Here we verify the effectiveness of considering other weather properties when we estimate a specific property, as mentioned in Section 5.4. Right now, the best average accuracy 0.596 (named W_0) is obtained by considering all weather-related features, with KPCA to reduce dimensions into 50, and with trimmed bagging. If we further consider (estimated) temperature (T_0) and humidity (H_0) values, the performance can be largely elevated to 0.766 (W_1) if we reduce the number of dimensions to 70, as shown in Fig. 10(b). This performance gain is impressive because we only further take two real values (temperature and humidity) into account. This shows high correlation between weather type and temperature/humidity, and may inspire many future weather property estimation studies.

Table 9
Average weather type classification accuracy obtained based on different types of features.

Weather-related features	CNN features (sky only)	CNN features (whole image)	Weather+CNN (sky)	Weather+CNN (whole)
0.583	0.495	0.473	0.577	0.591

Table 10 shows the confusion matrix of weather type classification, where columns show truth types and rows show estimated types. These results are very promising. Much more accurate results than random guess can be obtained by the proposed method. The relatively worse performance for rainy and cloudy photos is not beyond our expectation, and may attribute to noisy data and user's photo taking behavior. First, information of the meteorological station closest to a photo is used to be its ground truth. Sometimes weather conditions differ in two places even they are apart from each other by only four kilometers. Moreover, it is sometimes difficult for people to distinguish rainy photos from cloudy photos. Second, people tend to take photos with less rain even it is raining. We seldom see a photo taken on rainy days with many raindrops. In most cases, only gloomy sky can be seen in such photos.

6.2.5. The influence of imbalanced data

The experimental results shown above are all based on the selected data subset where the number of photos in each weather type is 1100 (1000 photos are for training, and 100 photos are for testing, Section 5.2.1). However, in Section 3.2 we mention that the numbers of photos in different weather types are imbalanced, i.e., most photos were taken on sunny or cloudy days. Here we would like to evaluate the influence of imbalanced data in weather type classification.

To make fair comparison, we randomly select 5500 photos from the Image2Weather dataset, where 68% is sunny, 22% is cloudy, around 1% is snowy, 7.5% is rainy, and around 1% is foggy. From this imbalanced subset, we randomly select 5000 photos for training, and the remaining 500 photos for testing. To make this experiment more convincing, we randomly build five imbalanced subsets of 5500 photos, and for each subset we run the aforementioned training-testing process 20 times. There are thus totally $5 \times 20 = 100$ training and testing processes. Table 11 shows the average confusion matrix obtained from the 100 experiments. Not surprisingly, because almost all training data are sunny or cloudy photos, the estimated class labels are mostly either sunny or cloudy. Overall, the average classification accuracy is 0.704. From this result, we see that the constructed random forest classifiers still provide better discrimination, comparing with that we always guess the class label as sunny (0.68). In the future, more elegant approaches specifically designed to handle imbalanced data can be investigated to further improve performance [45].

Table 10
The confusion matrix of weather type classification based on the random forest classifiers, with dimension reduction, trimmed bagging, and considering other weather properties.

	Sunny	Cloudy	Snowy	Rainy	Foggy
Sunny	0.83	0.09	0.01	0.02	0.05
Cloudy	0.06	0.67	0.05	0.18	0.04
Snowy	0.01	0.01	0.93	0.00	0.04
Rainy	0.02	0.18	0.02	0.68	0.11
Foggy	0.01	0.02	0.12	0.11	0.73
Average			0.766		

Table 11
The average confusion matrix of weather type classification based on imbalanced data subsets.

	Sunny	Cloudy	Snowy	Rainy	Foggy
Sunny	0.96	0.03	0.00	0.00	0.00
Cloudy	0.79	0.20	0.00	0.00	0.00
Snowy	0.80	0.18	0.01	0.00	0.00
Rainy	0.77	0.21	0.00	0.01	0.00
Foggy	0.76	0.24	0.00	0.00	0.00
Average			0.704		

6.3. Performance of temperature and humidity estimation

Similar to weather type classification, we perform temperature estimation for 100 runs (with the policies of dimension reduction, tree pruning, and considering other weather properties). At each run, 1000 photos are randomly selected as the training data, and the remaining 100 photos are for testing. The average Pearson correlation and Spearman correlation after 100 runs are 0.93 and 0.94, respectively. The estimation results are highly correlated with the ground truth. The average difference between the estimated temperature and the ground truth is around 1.98 °C.

For humidity estimation, after 100 runs, the average Pearson correlation and Spearman correlation are 0.85 and 0.80, respectively. The estimation results are also highly correlated with the ground truth. The average difference between the estimated humidity and the ground truth is around 7.13%. Generally, estimating humidity is relatively less reliable in our work.

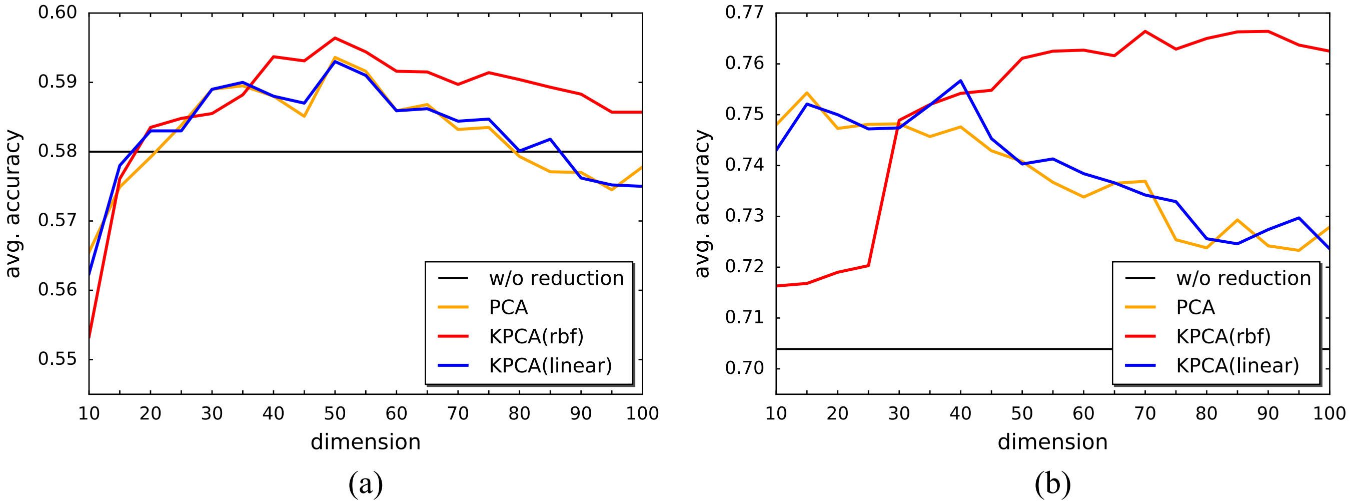


Fig. 10. Left: Average classification accuracy vs. dimensions of reduced feature vectors, without considering other weather properties. Right: Average classification accuracy vs. dimensions of reduced feature vectors, when considering other weather properties.

Fig. 11 shows some samples of weather type classification, temperature estimation, and humidity estimation. We see that promising results can be obtained if a photo was well taken to include the sky region.

6.4. Performance comparison

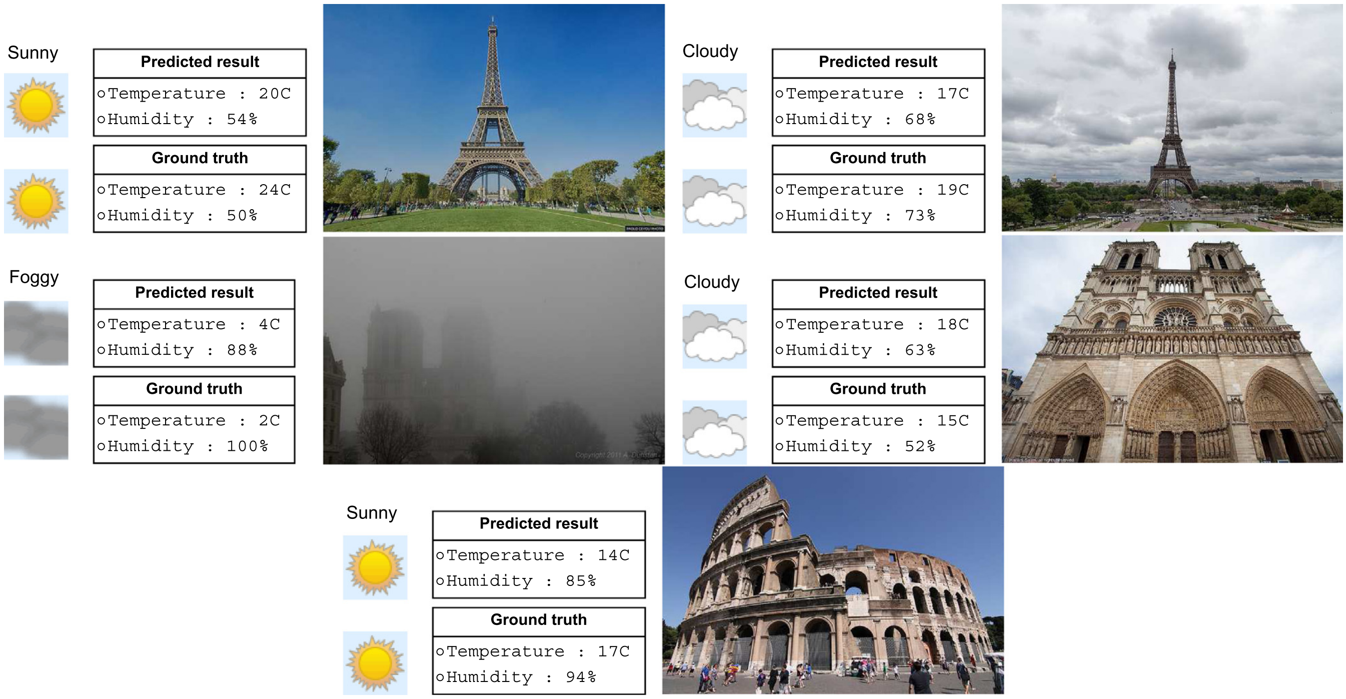
We compare the proposed weather type classification with the state-of-the-art work in [22]. To make comparison fair, we only consider two types of weathers, i.e., sunny and cloudy, and experiment on the dataset provided by [22], which consists of 5000 sunny and 5000 cloudy images. We extract seven types of features to represent images, exclusive of time information because timestamps of photos in this dataset are missing. In this experiment, 80% of the data are randomly selected as the training set, and the remaining is for testing. Five rounds of training and testing are conducted, and the average classification accuracy is reported. In this task, random guess reaches 50% accuracy. Therefore, in [22] a normalized accuracy value is designed to better express classification results. Particularly, the normalized accuracy is calculated as $max((a-0.5)/(1-0.5),0)$, where a is the accuracy obtained traditionally.

Table 12 shows accuracy and normalized accuracy of each weather type in the two-class image dataset, based on the proposed random forest classifier. Very promising performance can be obtained by the proposed features and classification method. Table 13 shows performance comparison between our method and [22], and several other variants mentioned in [22]. The SVM method builds SVM classifiers based on weather features mentioned in [22]. The Adaboost method takes each feature bin to build a weak classifier, and integrates all weak classifiers with a boosting strategy. The LLC method and the ScSPM method encode features based on locality-constrained linear coding and sparse coding with spatial pyramid matching, respectively. A component, sun visibility prediction, proposed in [21] is viewed as a weather classifier to build the Lalonde method. Two vehicle-based weather classifiers [46,14], i.e., images were all captured by cameras mounted on vehicles, are also compared. From this table we see that the state of the art [22] achieves significant performance improvement over all other methods just mentioned. By comparing ours with [22], we see another performance jump brought by the proposed features and the random forest classifier. Note that there are two differences between ours and [22]. First, in [22], a collaborative learning approach was proposed to integrate votes from different subsets of training data, while we integrate opinions derived from a large number of decision trees. Second, in [22], five types of features (sky, shadow, reflection, contrast, and haze) constituting a 621-D vector is used to represent an image, while we adopt seven types of features constituting a 562-D vector. The haze and contrast features we used are the same as [22]. Other features like color, texture, intensity, cloud, and LBP are extracted only from the sky region. In [22], only their "sky" features are extracted from the sky region.

Table 12
Classification accuracy for the two-class image dataset, based on the random forest classifier.

	Sunny	Cloudy	Average
Accuracy	0.86	0.94	0.90
Normalized accuracy	0.72	0.89	0.80

We are also interested in comparing the proposed method with the transient attributes estimation method proposed in [20]. They extracted histograms of oriented gradients (HOG), self-similarity features (SSIM), GIST, and geometric context color histograms as image features, and utilized support vector regression to estimate the score of each of 40 transient attributes. Of them, scores of five attributes including sunny, cloud, snow, rain, and fog are especially taken in comparison. Given an image from our balanced dataset, we estimate scores of these five transient attributes by the code released by the authors, and take the attribute with the highest score as the weather label of the given image. By evaluating all test images, we obtain 38.33% average classification accuracy, which is


Sunny
Predicted result
○Temperature : 20C
○Humidity : 54%
Ground truth
○Temperature : 24C
○Humidity : 50%

Cloudy
Predicted result
○Temperature : 17C
○Humidity : 68%
Ground truth
○Temperature : 19C
○Humidity : 73%

Foggy
Predicted result
○Temperature : 4C
○Humidity : 88%
Ground truth
○Temperature : 2C
○Humidity : 100%

Cloudy
Predicted result
○Temperature : 18C
○Humidity : 63%
Ground truth
○Temperature : 15C
○Humidity : 52%

Sunny
Predicted result
○Temperature : 14C
○Humidity : 85%
Ground truth
○Temperature : 17C
○Humidity : 94%

Fig. 11. Examples of weather property estimation. Photos at the top row are sunny and cloudy Eiffel Tower in Paris, respectively. Photos at the second row are foggy and cloudy Notre Dame in Paris, respectively. The photo at the last row is sunny Colosseum in Rome.

Table 13
Normalized accuracy (%) of different methods based on the two-class image dataset.

	SVM	Adaboost	LLC [47]	ScSPM [48]	Lalonde [21]	Yan [46]	Roser [14]	Lu [22]	Ours
Normalized accuracy	41.2	36.4	0.3	0.2	39.5	24.6	26.2	53.1	**80.0**

much lower than the proposed method yielding around 59% accuracy (without the two-stage framework). We observe that the method in [20] hardly can identify challenging rainy and foggy images.

7. Weather-aware landmark classification

Once we can estimate weather properties for any given image, a number of interesting applications can be built. For example, a customized weather map can be constructed to present finer weather properties. Because meteorological stations are not uniformly set up everywhere, resolution of meteorological phenomena monitoring is nonuniformly limited. With the proposed estimation model, every camera can be viewed as a weather sensor, and it is plausible to illustrate a weather map entirely constructed based on photos shared on the web. This idea becomes even more important if we keep track of the photos taken in remote districts, where only a limited number of equipments are set to monitor weather properties. The weather properties estimated from a large number of photos taken by tourists or mountain climbers, though not so accurate and professional, may be beneficial to climate change analysis.

In this paper, we take landmark classification as an instance to verify the benefit brought by weather information. We searched photos of twenty-five famous landmarks in Europe on Flickr. After filtering out photos with artificial editing and indoor photos, we finally collected a landmark dataset consisting of 13,973 images. The average number of photos per landmark is 558, while the most popular landmark (Colosseum in Rome) consists of 1404 photos, and the least popular landmark (Casa Mila in Barcelona) consists of 251 photos. Fig. 12 shows sample photos of each landmark. Following the single image classification process proposed in [49], we describe a photo by a bag of visual words (BoW) model, and then construct a multiclass SVM to achieve landmark classification. As a baseline system, we utilize the visual vocabulary (with 10,000 visual words) built in Top-SURF [50] to construct an image's BoW representation, and use the LibSVM package [51] to construct the multiclass SVM. To build a weather-aware landmark classification system, we simply concatenate the estimated weather type, temperature, and humidity to the end of the 10,000-D BoW representation, and also construct a multiclass SVM classifier.

Table 14
Average classification accuracy based on the collected landmark dataset.

	BoW	BoW+Weather	CNN	CNN+Weather
Avg. accuracy	46.90%	50.02%	89.02%	89.33%

Table 15
Average classification accuracy based on the landmark 3D dataset.

	BoW	BoW+Weather	CNN	CNN+Weather
Avg. accuracy	80.34%	80.59%	96.21%	96.30%

In all experiments, the random split scheme is adopted ten times for training and testing, and the average accuracy obtained from ten runs is reported. At each run, for each landmark 200 photos are randomly selected as the training data, and another 50 photos are randomly selected as the testing data. Table 14 shows performance of landmark classification based on BoW representation, BoW plus weather information, CNN features, and CNN plus weather information. By comparing the first two representations, we verify that weather information, though only in three dimensions, gives important clues and improve classification accuracy from 46.90% to 50.02%. Using CNN features yields 89.02% accuracy, and 89.33% accuracy can be obtained if weather information is further considered. Although the difference between CNN and CNN+Weather is marginal, it is statistically significant given that the p-value obtained by paired sample t test is around 0.0009. Two observations can be made from this result. First, this again confirms superiority of CNN features on image classification (landmark classification, in our case). Second, the benefit brought by weather information is more apparent when visual features are less descriptive, by comparing BoW+Weather with CNN +Weather. More performance gains may be expected if more weather information can be estimated and combined.

We also conduct the same procedure for the landmark 3D dataset [52]. There are also twenty-five landmarks in this dataset, and totally 45,180 photos are included. To make comparison fair, we also adopt the random split scheme five times for training and testing, and at each run, for each landmark 200 (50) photos are randomly selected as the training data (testing data). Table 15 shows average classification accuracy based on the landmark 3D dataset. By comparing Table 15 with Table 14, we can clearly see that

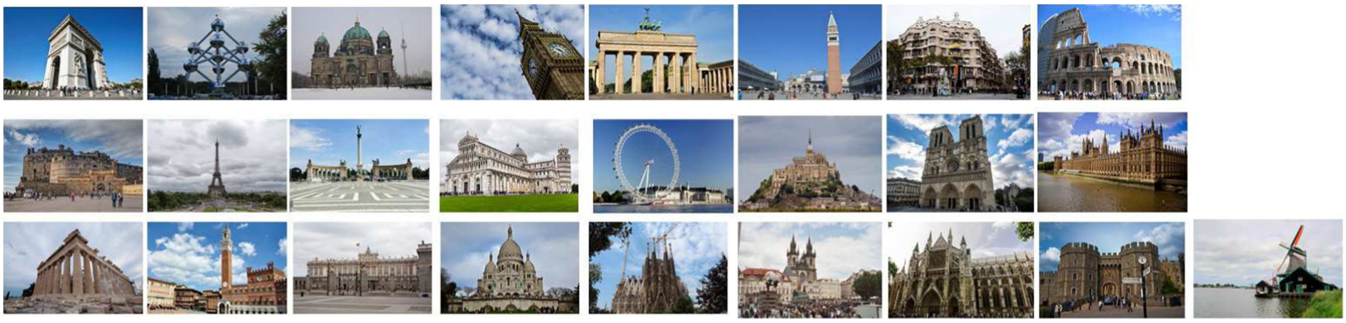


Fig. 12. Sample landmark photos. The landmarks from left to right, top to down, are Arc De Triomphe, Atomium, Berliner Dom, Big Ben, Brandenburg Gate, St Mark's Campanile, Casa Mila, Colosseum, Edinburgh Castle, Eiffel Tower, Heroes' Square, Leaning Tower of Pisa, London Eye, Mont Saint-Michel, Notre Dame, Palace of Westminster, Parthenon, Piazza del Campo, Royal Palace of Madrid, Basilica of the Sacred Heart of Paris, Sagrada Familia, Old Town Square, Westminster Abbey, Windsor Castle, and Zaanse Schans.

performance for the landmark 3D dataset is much higher than our dataset, even with the baseline BoW approach (80.34% accuracy). The landmark 3D dataset was mainly used to study 3D model construction from photos, and photos of the landmark may present less visual variation. The second observation from Table 15 is that, if data are relatively easier to be classified, adding weather information provides less benefit in classification. Though adding weather information just marginally improves classification accuracy, this effect is statistically significant (p-value = 0.001).

8. Summary

We have presented a systematic approach to estimate weather properties from single images. Our first contribution is building a large-scale image collection associated with various weather properties, image metadata, and elevation obtained from multiple platforms. As the second contribution, interesting statistics are given to show the relationships between weather properties and photo taking behaviors. Third, based on photo taken time and visual features, we build a weather type classifier based on the random forest approach. Temperature estimation and humidity estimation are respectively formulated as a regression problem and are solved by random forest regression. We describe potential applications and build a weather-aware landmark classifier to verify that weather information really aids computer vision applications.

Currently we only analyze three (weather type, temperature, humidity) of the various properties listed in Table 1. Based on the released large-scale image collection, we believe many potential applications can be built, and various future studies in both computer vision and human behavior analysis can be inspired. Moreover, currently only simple visual features and simple models are used to estimate weather properties. Specially designed features, or tags and comments given to photos, should be investigated with more elegant computational models to push this research field forward.

References

[1] K. van de Sande, T. Gevers, C. Snoek, Evaluating color descriptors for object and scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1582–1596.

[2] J. Wu, J. Rehg, Centrist: a visual descriptor for scene categorization, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1489–1501.

[3] N. Serrano, A. Savakis, J. Luo, A computationally efficient approach to indoor/outdoor classification, Proceedings of International Conference on Pattern Recognition, vol. 4, 2002, pp. 146–149.

[4] J. Hays, A. Efros, IM2GPS: Estimating geographic information from a single image, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[5] H. Katsura, J. Miura, M. Hild, Y. Shirai, A view-based outdoor navigation using object recognition robust to changes of weather and seasons, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 4, 2003, pp. 2974–2979.

[6] B.-F. Wu, C.-J. Chen, C.-C. Kao, C.-W. Chang, S.-T. Chiu, Embedded weather adaptive lane and vehicle detection system, in: Proceedings of IEEE International Symposium on Industrial Electronics, 2008, pp. 1255–1260.

[7] T.-L. Weng, Y.-Y. Wang, Z.-Y. Ho, Y.-N. Sun, Weather-adaptive flying target detection and tracking from infrared video sequences, Expert Syst. Appl. 37 (2010) 1666–1675.

[8] S. Divvala, D. Hoiem, J. Hays, A. Efros, M. Hebert, An empirical study of context in object detection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1271–1278.

[9] IPTC, Embedded Metadata Initiative. <http://www.embeddedmetadata.org/social-media-test-procedure.php>, 2016.

[10] C. Hauff, A study on the accuracy of Flickr's geotag data, in: Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval, 2013, pp. 1037–1040.

[11] S. Greengard, Weathering a new era of big data, Commun. ACM 57 (9) (2014) 12–14.

[12] S. Narasimhan, S. Nayar, Vision and atmosphere, Int. J. Comput. Vis. 48 (3) (2002) 233–254.

[13] S. Narasimhan, C. Wang, S. Nayer, All the images of outdoor scene, in: Proceedings of European Conference on Computer Vision, 2002, pp. 148–162.

[14] M. Roser, F. Moosmann, Classification of weather situations on single color images, in: Proceedings of IEEE Intelligent Vehicles Symposium, 2008, pp. 798–803.

[15] L. Shen, P. Tan, Photometric stereo and weather estimation using internet images, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1850–1857.

[16] Z. Chen, F. Yang, A. Lindner, G. Barrenetxea, M. Vetterli, How is the weather: automatic inference from images, in: Proceedings of IEEE International Conference on Image Processing, 2012, pp. 1853–1856.

[17] N. Jacobs, N. Roman, R. Pless, Consistent temporal variations in many outdoor scenes, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[18] N. Jacobs, W. Burgin, R. Speyer, D. Ross, R. Pless, Adventures in archiving and using three years of webcam images, in: Proceedings of IEEE CVPR Workshop on Internet Vision, 2009, pp. 39–46.

[19] M. Islam, N. Jacobs, H. Wu, R. Souvenir, Images+weather: collection, validation, and refinement, in: Proceedings of IEEE CVPR Workshop on Ground Truth, 2013.

[20] P. Laffont, Z. Ren, X. Tao, C. Qian, J. Hays, Transient attributes for high-level understanding and editing of outdoor scenes, ACM Trans. Graph. (Proc. SIGGRAPH 2014 33 (4) (2014).

[21] J.-F. Lalonde, A. Efros, S. Narasimhan, Webcam clip art: appearance and illuminant transfer from time-lapse sequences, ACM Trans. Graph. (Proc. SIGGRAPH Asia 2009 28 (5) (2009).

[22] C. Lu, D. Lin, J. Jia, C.-K. Tang, Two-class weather classification, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3718–3725.

[23] J. Xiao, J. Hays, K. Ehinger, A. Oliva, A. Torralba, Sun database: large-scale scene recognition from abbey to zoo, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3485–3492.

[24] Z. Zhang, H. Ma, H. Fu, C. Zhang, Scene-free multi-class weather classification on single images, Neurocomputing 207 (2016) 365–373.

[25] Z. Zhang, H. Ma, H. Fu, L. Liu, C. Zhang, Outdoor air quality level inference via surveillance cameras, Mob. Inform. Syst. 2016 (2016).

[26] D. Glasner, P. Fua, T. Zickler, L. Zelnik-Manor, Hot or not: exploring correlations between appearance and temperature, in: Proceedings of IEEE International Conference on Computer Vision, 2015, pp. 3997–4005.

[27] A. Volokitin, R. Timofte, L.V. Gool, Deep features or not: temperature and time prediction in outdoor scenes, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition Workshop, 2016, pp. 63–71.

[28] Y. Avrithis, Y. Kalantidis, G. Tolias, E. Spyrou, Retrieving landmark and non-landmark images from community photo collections, in: Proceedings of ACM International Conference on Multimedia, 2010.

[29] A. Vedaldi, K. Lenc, Matconvnet – convolutional neural networks for matlab, in: Proceedings of ACM International Conference on Multimedia, 2015.

[30] A. Quattoni, A. Torralba, Recognizing indoor scenes, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[31] B. Manjunath, W. Ma, Texture features for browsing and retrieval of image data, IEEE Trans. Pattern Anal. Mach. Intell. 18 (8) (1996) 837–842.

[32] Q. Li, W. Lu, J. Tang, J. Wang, Thin cloud detection and all-sky images using markov random fields, IEEE Geosci. Rem. Sens. Lett. 9 (3) (2012) 417–421.

[33] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.

[34] H.-M. Wu, Y.-J. Tien, C.-H. Chen, Gap: a graphical environment for matrix visualization and cluster analysis, Comput. Stat. Data Anal. 54 (2010) 767–778.

[35] K. He, J. Sun, X. Tang, Single image haze removal using dark channel prior, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1956–1963.

[36] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

[37] L. Breiman, J. Friedman, R. Plshen, C. Stone, Classification and Regression Trees, Champan and Hall/CRC, 1984.

[38] T.-E. Tseng, W.-Y. Chang, C.-S. Chen, Y.-C.F. Wang, Style retrieval from natural images, in: Proceedings of IEEE Conference on Acoustics, Speech and Signal Processing, 2016.

[39] H. Fu, Q. Zhang, G. Qiu, Random forest for image annotation, in: Proceedings of European Conference on Computer Vision, 2012.

[40] C. Crouxa, K. Joossensa, A. Lemmens, Trimmed bagging, Comput. Stat. Data Anal. 52 (2007).

[41] R. Mihail, S. Workman, Z. Bessinger, N. Jacobs, Sky segmentation in the wild: an empirical study, in: Proceedings of IEEE Winter Conference on Applications of Computer Vision, 2016.

[42] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[43] A. Krizhevsky, I. Sutskever, G. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of Advances in Neural Information Processing System, 2012.

[44] A. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, CNN features off-the-shelf: an astounding baseline for recognition, in: Proceedings of CVPR Workshop on DeepVision, 2014.

[45] H. He, E. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.

[46] X. Yan, Y. Luo, X. Zheng, Weather recognition based on images captured by vision system in vehicle, in: Proceedings of International Symposium on Neural Networks, 2009, pp. 390–398.

[47] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2010.

[48] J. Yang, K. Yu, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[49] Y. Li, D. Crandall, D. Huttenlocher, Landmark classification in large-scale image collections, in: Proceedings of IEEE Conference on Computer Vision, 2009, pp. 1957–1964.

[50] B. Thomee, E. Bakker, M. Lew, TOP-SURF: a visual words toolkit, in: Proceedings of ACM International Conference on Image and Video Retrieval, 2010, pp. 1473–1476.

[51] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011).

[52] Q. Hao, R. Cai, Z. Li, L. Zhang, Y. Pang, F. Wu, 3d visual phrases for landmark recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3594–3601.