

RSCM: Region Selection and Concurrency Model for Multi-Class Weather Recognition

Di Lin, Cewu Lu, *Member, IEEE*, Hui Huang, *Member, IEEE*, and Jiaya Jia, *Senior Member, IEEE*

Abstract—Toward weather condition recognition, we emphasize the importance of regional cues in this paper and address a few important problems regarding appropriate representation, its differentiation among regions, and weather-condition feature construction. Our major contribution is, first, to construct a multi-class benchmark data set containing 65 000 images from six common categories for sunny, cloudy, rainy, snowy, haze, and thunder weather. This data set also benefits weather classification and attribute recognition. Second, we propose a deep learning framework named region selection and concurrency model (RSCM) to help discover regional properties and concurrency. We evaluate RSCM on our multi-class benchmark data and another public data set for weather recognition.

Index Terms—Deep learning, multi-class weather recognition, image classification, attribute recognition.

I. INTRODUCTION

IMAGE weather recognition is a relatively new topic in computer vision [1]–[4]. Different from other object/scene recognition problems [5]–[8], weather recognition needs to understand complex phenomena of lighting and reflection on object surface and of the scene.

Weather is an integral part of our lives. Instantly and densely collecting weather information is a scientific topic with enormous social impact. Images carry a lot of information to understand weather. Images also have the special advantages in terms of large quantity for weather analysis and low cost to generate them.

Image-based weather recognition finds many applications. One example is that smart power grids, including solar energy

Manuscript received August 3, 2016; revised January 1, 2017 and March 10, 2017; accepted April 5, 2017. Date of publication April 19, 2017; date of current version June 23, 2017. This work is supported in part by the Research Grants Council of the Hong Kong SAR under Project 413113, in part by the National Natural Science Foundation of China under Project 61472245, in part by the NSFC under Grant 61522213, in part by 973 Program under Grant 2015CB352501, in part by the Guangdong Science and Technology Program under Grant 2015A030312015 and Grant 2016A050503036, Shenzhen Innovation Program under Grant JCYJ20151015151249564, and in part by the Natural Science Foundation of SZU under Grant 827-000196. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Guoliang Fan. (*Corresponding author: Cewu Lu.*)

D. Lin and H. Huang are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: ande.lin1988@gmail.com; hhzhijian@gmail.com).

C. Lu is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Minhang Qu 200240, China (e-mail: lu-cw@cs.sjtu.edu.cn).

J. Jia is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2017.2695883

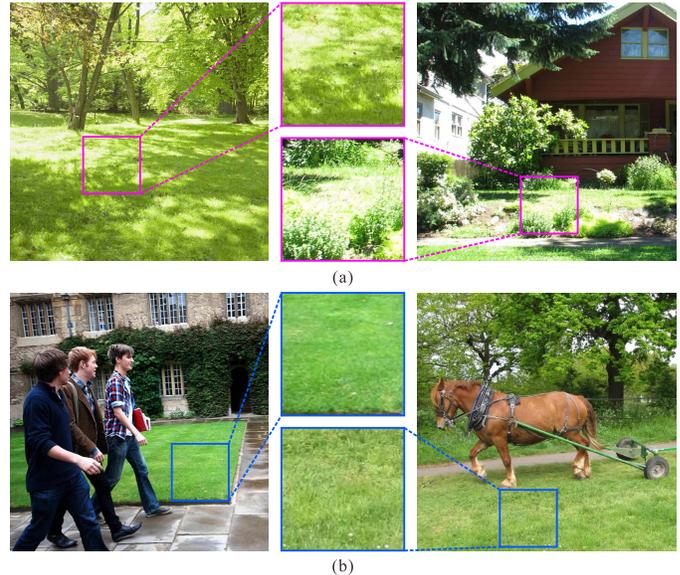


Fig. 1. Regions of meadow in sunny and cloudy days in (a) and (b) respectively.

system, can dispatch electric power according to the result of weather recognition. Driving assistance can also benefit from the instant response, as reported in [1]. Image weather recognition also helps develop outdoor robotics [9]. In computer vision, image-based weather recognition profits a group of research regarding scene understanding and image retrieval.

Despite useful, weather recognition from a single image is still challenging. The difficulty stems from the unique properties of weather, which make many global and local invariant features effective in object recognition and classification unusable [3]. Weather recognition methods [5], [10], [11] are also notably different from scene recognition because variation of scenes in various meteorological conditions, instead of scene structure itself [12], needs to be analyzed.

In this paper, we recognize weather condition via important regions and propose the new coexistence clue to determine a weather situation from single images. For example, blue sky with cast shadow on street together strongly suggest a sunny day. Our finding is briefly described below.

A. Regional Cue and Concurrency Condition

Natural images contain clusters of region context, e.g., “building”, “road” and “sky”. We observe reasonably stable weather patterns for these clusters. Figure 1 shows

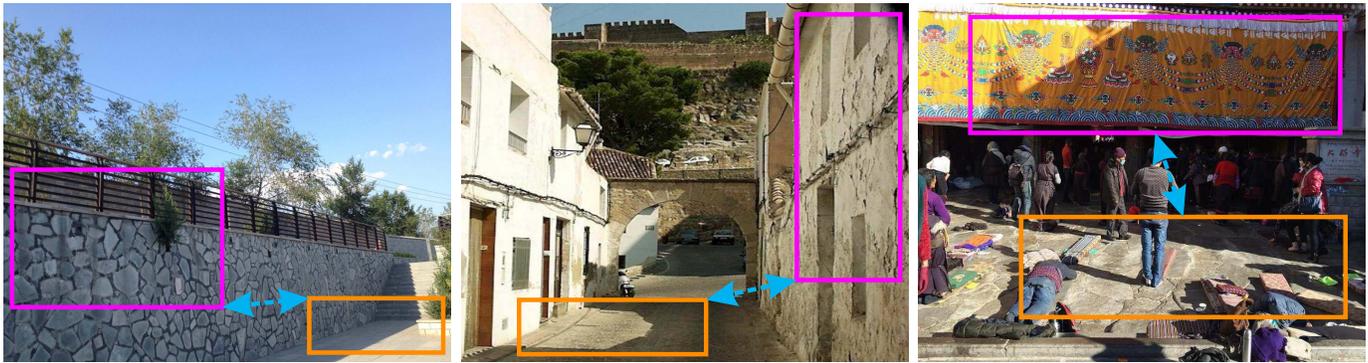


Fig. 2. Region concurrency for building & street. The street is partly lit with cast shadow of the wall. This type of concurrency becomes a type of feature in “sunny days”.

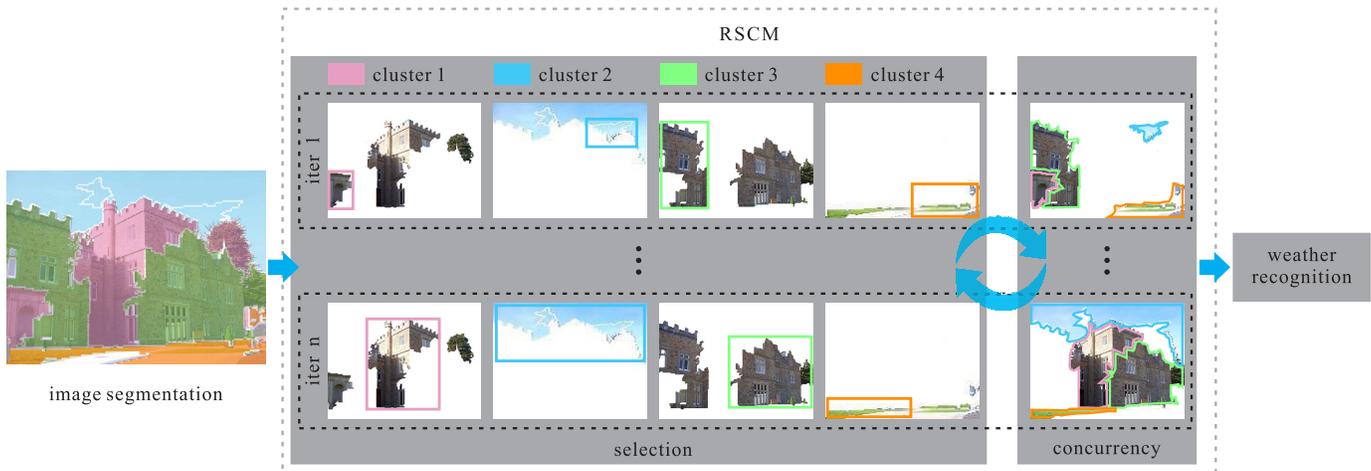


Fig. 3. Region selection and concurrency of an image. The input image is partitioned into 4 region clusters in this example. The middle illustrates RSCM modeled as an iterative process. In each iteration, the *selection* module scores regions from different clusters. The *concurrency* module evaluates co-existence of each region pair. These two components cooperate to iteratively update the model. The final regions are used for weather recognition. In this figure, the regions are only for the illustration purpose and we actually use regional features to represent them in experiments.

meadow examples, which can be differentiated between the sunny and cloudy classes based on their visual appearance.

Besides regional patterns, combination of regions from different clusters indicates high concurrency. These region pairs provide useful evidence to recognize weather situations. Figure 2 shows an example that the building casts shadow on the neighboring street under sunlight. This concurrency is a common phenomenon in many scenes, detailed later in this paper.

B. Difficulties

Object/scene recognition [13], [14] does not sufficiently explore weather-related regions. In order to construct image representation, previous methods focus on extracting as much common structural information as possible. Our first challenge is to select suitable regions for weather recognition.

For the concurrency condition, region pairs with high visual concurrency do not necessarily hold stable spatial/structural relationship as shown in Figure 2, unlike those in object/scene recognition [15], [16]. The concurrency condition could be rather sensitive to weather factors, such as direction of sunlight or reflection. It is therefore critical to propose a suitable

strategy to find concurrency appearing frequently in natural images.

C. Our Contribution

We propose a region selection and concurrency model (RSCM) to tackle these difficulties. Our RSCM is based on deep convolutional neural network (CNN). Figure 3 demonstrates the work flow of our RSCM with two major components – region selection and concurrency processes. For pre-processing, we segment an image into regions. For each region cluster, a latent variable is used for indicating the region that provides appropriate weather discrimination information. Instead of only considering neighboring regions, we discover visual concurrency on all region pairs. Their estimation is accomplished via the Siamese architecture of neural networks [17], [18] of RSCM, which powerfully handles the complex combination of regional patterns.

Our RSCM also makes it possible to train a weakly supervised framework with only image weather labels. It facilitates automatic discovery of regional weather cues and concurrency in a data-driven manner, which is hard to be defined beforehand.

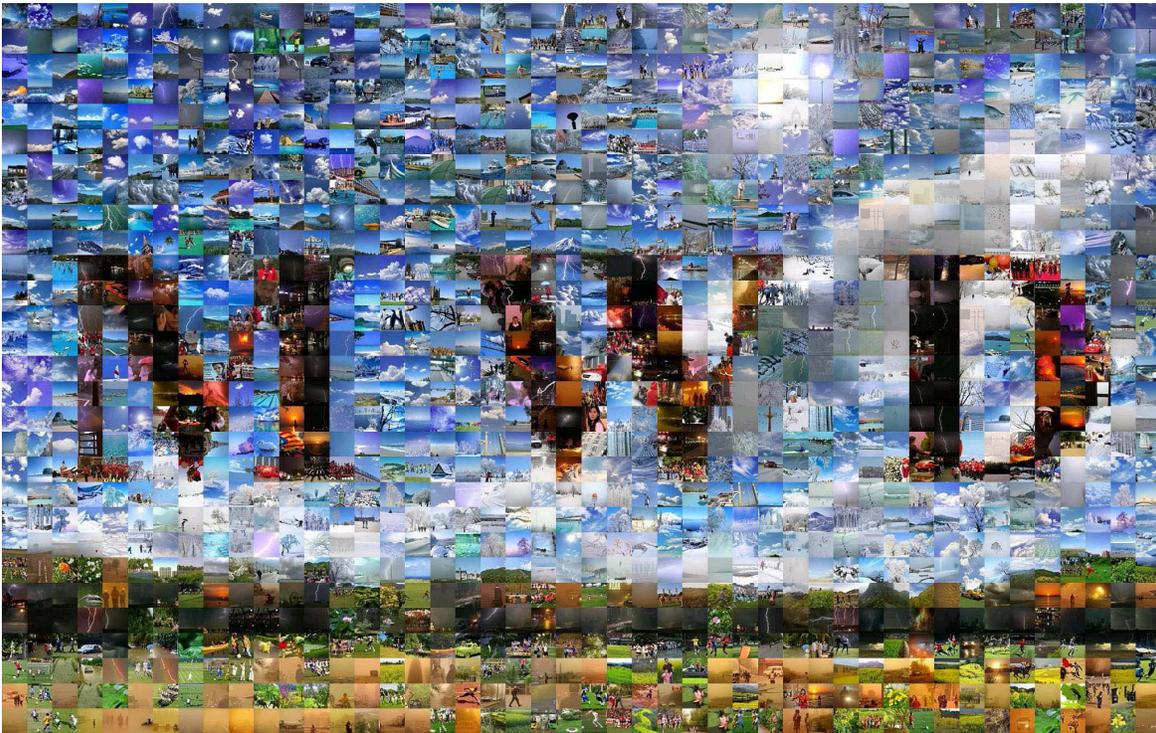


Fig. 4. Overview of our multi-class weather dataset. It includes 6 weather categories, i.e., *sunny*, *cloudy*, *rainy*, *snowy*, *haze* and *thunder*.

In addition, we construct a dataset with 65,000 images from 6 common weather categories, readily usable for multi-class weather recognition. We use our data as benchmark for weather classification and attribute recognition. Compared with previous work, our RSCM yields decent performance in terms of multi-class classification and attribute prediction. We also conduct comprehensive study to show weather information is transferable to attribute recognition.

II. RELATED WORK

There are works to construct a weather recognition algorithm for driving assistance. Yan et al. [1] proposed detecting regions of interest based on road information, which was captured by vehicle cameras. Roser et al. [2] followed the traditional bag-of-feature strategy considering brightness, contrast, sharpness, saturation and hue in sunny and rainy days. A histogram was constructed as the whole-image representation for weather classification. These methods need special hardware to recognize weather situation.

Recently, research on weather classification focuses on adding weather priors. Lu et al. [3] presented a framework for single natural images. Sky, shadow and haze information was used to construct weather-specific features. The voting models were employed to incorporate all features into the inference and learning framework. Recently, Zhang et al. [4] extended the similar framework to multi-class weather classification. They designed features for sunny, rainy, snowy and haze weather. With the pre-defined features, multi-kernel learning was employed to recognize weather condition.

Several related methods estimate illumination from outdoor images. Lalonde et al. [19] utilized sky, shadow on the ground

and shading on vertical surface to compute the distribution over the sun position and visibility. Kim et al. [20] took a sequence of images recorded by a fixed camera to compute illumination change. Wehrwein et al. [21] estimated the direction of sun and detected shadow from a set of images.

These methods do not consider variant patterns of distinct region clusters and their concurrence. Our region selection and concurrency is new for solving this problem in another line. Our method makes use of deep learning for the new recognition task. Besides, We contribute a larger multi-class weather dataset, allowing further analysis and training reliable systems.

III. MULTI-CLASS WEATHER DATASET

Our newly constructed multi-class weather dataset (MWD) contains 65,000 images from 6 common categories, i.e., *sunny*, *cloudy*, *rainy*, *snowy*, *haze* and *thunder* weather, as shown in Figure 4.

A. Previous Datasets

Publicly available datasets for weather classification are limited in scale and quantity. The dataset provided in [2] includes weather conditions of *clear*, *light rain*, and *heavy rain*. These images are rain-oriented, obtained specially from expressway surveillance. Lu et al. [3] provided a weather classification dataset with more natural scenes. But only sunny and cloudy images are involved. Zhang et al. [4] described a multi-class weather dataset containing four classes of weather conditions. In comparison to previous datasets, ours comprises of more images with different weather classes.



Fig. 5. Sample images of our (a) weather classification dataset and (b) weather attribute dataset.

B. Data Collection

To reduce bias during data collection and annotation, we asked 12 annotators to build the dataset. All annotators were unaware of the methods used in future weather experiments. To determine the categories in our dataset, the 12 annotators collected images from Flickr and Google using keywords “outdoor” and “weather”. Around 10,000 images were initially gathered and annotated. Eventually, we select the six most general weather categories to form our dataset.

Then annotators were asked to collect more images within these six categories. Only images containing photo-realistic outdoor scenes and with reasonable resolutions were kept. Visually similar images were rejected by computing the color histogram distances for all image pairs, similar to what was done in [3]. After this step, we have around 12,000 images for each category.

C. Annotation

The images are divided into two groups. They are further annotated with weather classes and attributes.

1) *Weather Class Annotation*: We make a group of our dataset used for weather classification, where the critical weather condition of an image is identified. Although only a single weather class per image is labeled, the classification data helps discover and analyze characteristics of each weather condition. The properties can be combined flexibly to study their correlation.

Weather conditions may co-exist. It is thus ambiguous to give each image one category label. To tackle it, we classify the images as “normal” ones if they do not contain apparent rain drop, snow or haze as shown in Figure 6(a). These images are labeled as sunny or cloudy. Other images with heavy rain, snow and thunder, as shown in Figure 6(b), are assigned to “bad” weather condition.¹

During annotation, all annotators knew our discipline. The 12 annotators were divided into 6 groups. The two annotators in each group independently check the images in corresponding categories. Images labeled as ambiguous weather by either of them were discarded. Finally, we maintain

¹The normal and bad weather is classified by the condition of [2]. We extend the differentiation criteria using rainy, snowy, haze and thunder conditions together.

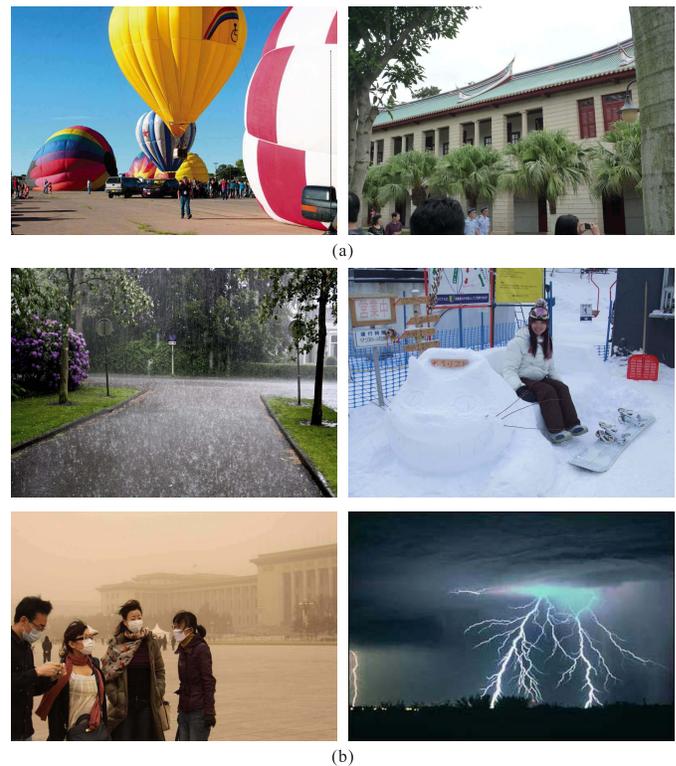


Fig. 6. Sample images of (a) clear weather conditions and (b) bad weather conditions.

10,000 images for each weather category. Sample images are shown in Figure 5(a).

2) *Weather Attribute Annotation*: The other group of data is annotated to support weather attribute recognition, which aims to find all weather conditions in an image. Comparing with classification, attribute recognition allows more flexible weather recognition.

During attribute annotation, we make use of selected 12,000 images. These images are likely to have two or more weather conditions since they cause ambiguity during class annotation. About 2,000 images were assigned to each 2 annotators. Again, the 2 annotators independently labeled the images with attributes. After labeling, an extra process was applied to check each image labeled by the 2 annotators.

If an image was labeled with at least one ambiguous attribute, it is eliminated. Totally we have 5,000 images with weather attributes. Samples are shown in Figure 5(b).

To the best of our knowledge, weather attribute data was not provided by any other publicly available datasets. We believe ours will profit related research.

IV. OUR FRAMEWORK

Our framework to recognize weather is illustrated in Figure 3. First, we segment the image into different regions. For each of them, we extract regional features for representation. We apply these features and the region selection and concurrency model (RSCM) to weather recognition. Without loss of generality, we elaborate on our RSCM for binary classification where each label is denoted as $y = \{-1, 1\}$. We then extend this framework to multi-class classification and attribute recognition by training independent RSCM for each weather class.

A. Pre-Process: Coarse Segmentation

Outdoor images are composed of regions. Figure 8(a) shows an example with four clusters of regions. Regions in each cluster are similar in appearance. Image semantic labeling techniques [22]–[25] aim to find these region clusters under supervised learning with human annotation. The output regions are with semantic categories, e.g., building and tree. Since we only need to know whether two regions come from the same cluster or not, co-segmentation [26]–[28] is adopted in our framework, which works in an unsupervised manner to segment an image into different regions.

The resulting clusters each contains a few separate regions. Figure 8(b) shows the segmentation result using the method of [26] where cluster 1 involves 2 regions marked in green. Note our method does *not* require regions coming from semantic classes or accurate in terms of class labels. Coarsely segmented regions without semantic classes are enough for our weather recognition.

Regarding possible variant region clusters mistaken by co-segmentation [26]–[28], we apply the K-means algorithm to group images based on global appearance features [10], [29]. With this step, each group of images are close in appearance, as shown in Figure 7. The generated region clusters are illustrated in Figure 9.

After K-means to partition images, for each group we train a co-segmentation model. During training and testing, each image is processed by corresponding co-segmentation. We eliminate regions with size smaller than 200 pixels. More details are given in Section VII.

B. Regional Feature Extraction

We use convolutional feature masking (CFM) [30] to extract regional feature of the segmented regions. CFM is a fast way to extract feature of irregular regions on the convolutional feature map. Figure 10 illustrates the pipeline. We denote \mathcal{R} as a region in image \mathbf{I} and compute the convolutional maps on the image \mathbf{I} . Then we keep the down-sample ratio to map region \mathcal{R} from image domain to the convolutional map.



Fig. 7. Examples of image groups generated by k-means. Each group of images have similar appearances.

We apply region-of-interest (RoI) pooling [31] and masking to generate the fixed-length feature. This fixed-length feature is processed with fully-connected layers to form the final regional feature. We denote the regional feature of \mathcal{R} as

$$\mathbf{f} = G(\Theta; \mathbf{I}, \mathcal{R}), \quad (1)$$

where the function $G(\Theta; \cdot)$ represents the architecture of CNN. Θ is the model parameter.

C. Region Selection and Concurrency

Region clusters are illustrated in Figure 11(a) along with their features. To recognize weather, we further address two problems. The first is region selection for each cluster. In Figure 11(b), the 1st cluster contains two regions and only the bottom region contains pattern of shadow, which is important for understanding it as a sunny day. The other region is not similarly useful. How to find and give importance to respective regions is a challenge.

The second problem is region concurrency. Some regions from different clusters should be considered simultaneously as exemplified in Figure 11(b) and (c) where the two highlighted regions form causality under sun light regarding the cast shadow. Thus concurrency frequently yielded in training data should be considered, which can make weather recognition more reliable. We propose RSCM to tackle these two problems.

1) *Region Selection*: Given an image \mathbf{I} , we denote $\mathcal{R}_{i,j,k}$ as the i^{th} region in the j^{th} cluster of the k^{th} image group. We note the image group k is determined by K-means clustering and is regarded as a constant. For each k , we set the regional features computed by other image group models as zero vectors, which do not affect training and inference. For simplicity, we omit subscript k and use notation $\mathcal{R}_{i,j}$.

The feature of $\mathcal{R}_{i,j}$ is denoted as $\mathbf{f}_{i,j}$. The number of regions of the j^{th} cluster is denoted as n_j . We introduce the binary variable $v_{i,j}$ with value 1 (or 0) to denote that region $\mathcal{R}_{i,j}$ is (or is not) selected. Our RSCM selects one region for each cluster. We define $\mathbf{f}_{0,j} = \mathbf{0}$ as a zero vector and let $v_{0,j} = 1$



Fig. 8. (a) Image with regions containing building, road, tree and sky. (b) Partitioning of the image by co-segmentation. Each cluster contains a few separate regions.

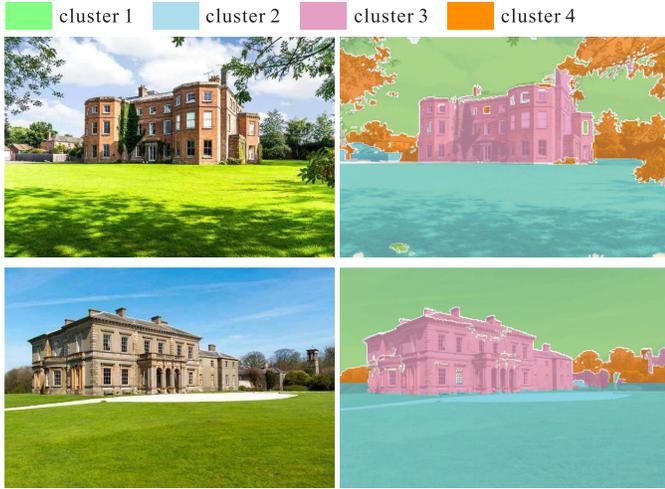


Fig. 9. Two images from the same group. Their co-segmentation results show similar appearances that can be reasonably divided into stable region clusters.

only when $n_j = 0$. This setting allows us to select a zero vector from a cluster, which has no visible region. The selected feature is formulated as

$$\mathbf{F}_j = v_j^T \mathbf{f}_j, \quad (2)$$

where

$$v_j = \begin{bmatrix} v_{0,j} \\ \vdots \\ v_{n_j,j} \end{bmatrix}, \quad v_{i,j} \in \{0, 1\}, \quad \|v_j\|_1 = 1, \quad \mathbf{f}_j = \begin{bmatrix} \mathbf{f}_{0,j} \\ \vdots \\ \mathbf{f}_{n_j,j} \end{bmatrix}, \quad (3)$$

$v_j \in \mathbb{R}^{n_j+1}$ and $\mathbf{f}_j \in \mathbb{R}^{(n_j+1) \times D}$. $\mathbf{F}_j \in \mathbb{R}^D$ is the selected feature from the j^{th} cluster. D is the dimension of the feature, which is 4,096 in our experiment. Our region selection of an image is formulated as

$$E_s(\mathbf{W}^s; \mathbf{V}, \mathbf{F}) = \text{trace}(\mathbf{W}^s \mathbf{F}^T), \quad (4)$$

where

$$\mathbf{W}^s = \begin{bmatrix} \mathbf{w}_1^s \\ \vdots \\ \mathbf{w}_M^s \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} v_1 \\ \vdots \\ v_M \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \mathbf{F}_1 \\ \vdots \\ \mathbf{F}_M \end{bmatrix}, \quad (5)$$

$\mathbf{W}^s \in \mathbb{R}^{M \times D}$ is the model parameter. \mathbf{V} contains all binary variables. $\mathbf{F} \in \mathbb{R}^{M \times D}$ contains all the selected features from different region clusters. The model parameter \mathbf{w}_j^s is used to weight the selected feature \mathbf{F}_j from the j^{th} cluster. Their dot product is regarded as a score that predicts an image to have a positive label, i.e., $y = 1$. As in Eq. (4), we sum all the scores as an overall confidence for the region selection process. The weather condition of an image can be classified by a set of discriminative regions. We formulate Eq. (4) as a scoring function. It outputs high score to encourage the selection of discriminative regions.

2) *Region Concurrency*: We measure the concurrency of two regions in different clusters using a Siamese architecture of deep neural networks. Figure 10 illustrates our Siamese network. Following [17], [18], we model this network as a two-branch architecture. These two branches share the same network parameters of all convolutional and fully connected layers. During training and testing, the Siamese network accepts a pair of regions from different clusters as input. These two regions are used by CFM to generate distinct regional features. This design accounts for the complexity of concurrency patterns.

Using the regional features, the Siamese network outputs the concurrency of a pair of regions in the j^{th} and k^{th} classes. The concurrency is computed as

$$s_{j,k} = -\|\mathbf{F}_j - \mathbf{F}_k\|_2. \quad (6)$$

Larger values of $s_{j,k}$ refer to higher concurrency of selected regions.

Given M clusters, we compute concurrency of each pair of regions from different clusters. Using Eq. (6), we formulate region concurrency as

$$E_c(\mathbf{W}^c; \mathbf{V}, \mathbf{F}) = \sum_{j=1}^{M-1} \sum_{k>j}^M (\mathbf{W}^c \circ \mathbf{S})_{j,k}, \quad (7)$$

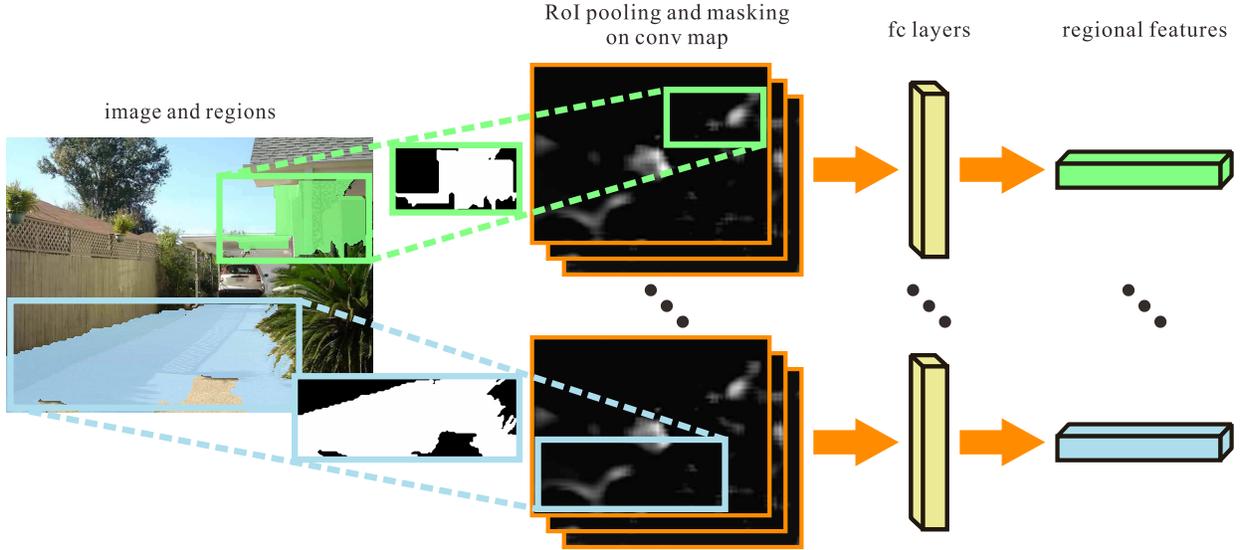


Fig. 10. Architecture of the Siamese CNN for regional feature extraction. Input contains an image and segmented regions. All region-of-interests (RoIs) of the input image share the identical convolutional map and fully-connected layers.

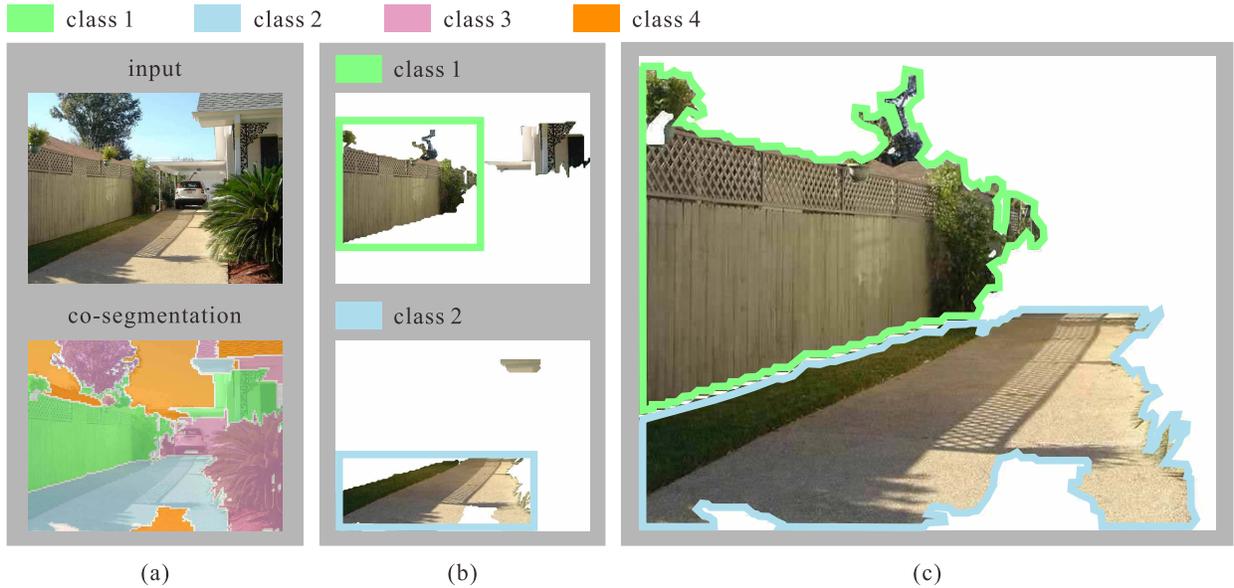


Fig. 11. (a) Image is segmented into regions. (b) Regions in the 1st and 2nd clusters. (c) The 1st and 2nd regions are with high concurrency for indication of sunny weather. We actually use regional features to represent the regions in experiments.

where $\mathbf{W}^c, \mathbf{S} \in \mathbb{R}^{M \times M}$. Operator \circ denotes Hadamard product. As the concurrency matrix \mathbf{S} is symmetric, i.e., $s_{j,k} = s_{k,j}$, the matrix \mathbf{W}^c is also symmetric for consistency. The element $w_{j,k}^c \in \mathbf{W}^c$ is a weight for the concurrency $s_{j,k} \in \mathbf{S}$. The constraint $k > j$ means we measure the concurrency of regions from different clusters. Scores of all concurrency are added to form overall confidence. The parameter \mathbf{W}^c is learned from the training data. Compared to separate regions without any relationship, regions that stably co-exist provide more useful information for weather recognition. As in Eq. (7), the region concurrency is also modeled as a scoring function. A higher score of Eq. (7) means the regions have higher concurrency.

3) *Joint Model for Weather Recognition*: Since region selection and concurrency work together, we combine them to form

the decision function of our RSCM. With Eqs. (4)-(7), the decision function is expressed as

$$J(\mathbf{W}^s, \mathbf{W}^c; \mathbf{V}, \mathbf{F}) = E_s(\mathbf{W}^s; \mathbf{V}, \mathbf{F}) + E_c(\mathbf{W}^c; \mathbf{V}, \mathbf{F}). \quad (8)$$

The decision function Eq. (8) assigns a score by balancing region selection and concurrency. The discriminative regions that also have stable concurrency pattern lead to higher score of Eq. (8). A higher score means that an image has higher confidence to have label $y = 1$.

V. SYSTEM LEARNING

During the learning phase, we are given a training set \mathcal{T} . We denote $(\mathbf{F}, y) \in \mathcal{T}$ as a training sample, where \mathbf{F} is the set of regional features and y is the weather label of an image.

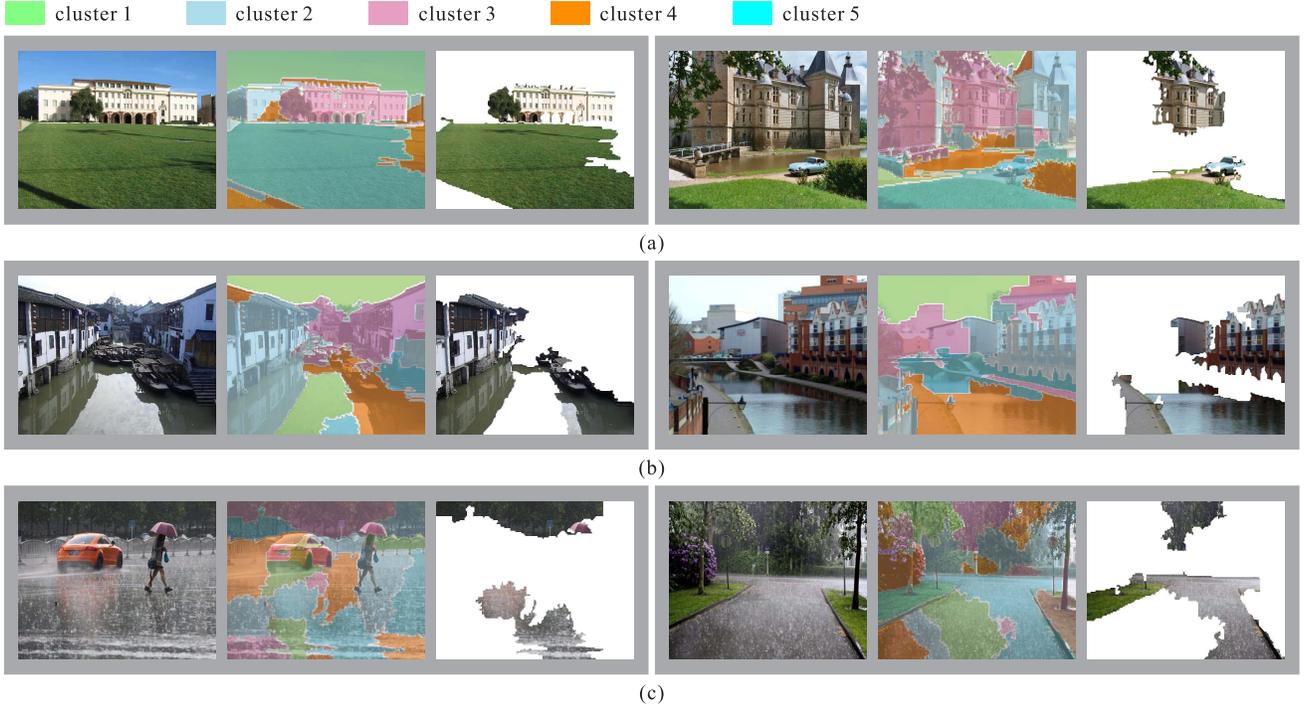


Fig. 12. Inferred region pairs, which indicate (a) sunny, (b) cloudy and (c) rainy days. The regions are represented as regional features in experiments.

The learning task is to determine the model parameters \mathbf{W}^s , \mathbf{W}^c and Θ . We formulate learning of \mathbf{W}^s , \mathbf{W}^c and Θ in iterative optimization.

A. Inference of \mathbf{V}

The latent variable \mathbf{V} indicates the selected/unselected regions, which should be determined before optimization of the model parameters. We optimize Eq. (8) over the latent variable \mathbf{V} as

$$\mathbf{V}^* = \arg \max_{\mathbf{V}} J(\mathbf{W}^s, \mathbf{W}^c; \mathbf{V}, \mathbf{F}). \quad (9)$$

According to the definition in Section IV-C.1, \mathbf{V} is a discrete variable. Although the solution space is limited, exhaustive search is not recommended for large M .

We turn to iterated conditional modes (ICM) [32] for solving for \mathbf{V} . ICM is an iterated algorithm. We randomly initialize \mathbf{V} . In each iteration, we update the set of items $\{v_{i,j} | j = m\}$ conditioned on the fixed set $\{v_{i,j} | j \neq m\}$. It means that we update the states of a region cluster in one iteration. The same update also applies to other region clusters and lasts for iterations, until Eq. (8) converges. Figure 12 shows examples of the inferred region pairs.

B. Learning of \mathbf{W}^s and \mathbf{W}^c

For the training sample (\mathbf{F}, y) , we use Eq. (9) to fix latent variable \mathbf{V} . Following latent SVM [15], learning \mathbf{W}^s and \mathbf{W}^c is accomplished by minimizing the following objective function

over $\mathbf{W} = [\mathbf{W}^s, \mathbf{W}^c]$.

$$\begin{aligned} L(\mathbf{W}; \mathcal{T}, \mathbf{V}) &= \frac{1}{2} \|\mathbf{W}\|_2^2 + \gamma \sum_{(\mathbf{F}, y) \in \mathcal{T}} \max(0, 1 - y\mathbf{W} \cdot \Phi), \\ s.t. \quad \mathbf{W} &= [\mathbf{w}_1^s, \dots, \mathbf{w}_M^s, w_{1,2}^c, w_{1,3}^c, \dots, w_{M-2,M}^c, w_{M-1,M}^c], \\ \Phi &= [\mathbf{F}_1, \dots, \mathbf{F}_M, s_{1,2}, s_{1,3}, \dots, s_{M-2,M}, s_{M-1,M}], \end{aligned} \quad (10)$$

where γ is set to 0.07 empirically. We apply gradient descent where the gradient w.r.t. \mathbf{W} is computed as

$$\frac{\partial L(\mathbf{W}; \mathcal{T}, \mathbf{V})}{\partial \mathbf{W}} = \mathbf{W} - \gamma \sum_{(\mathbf{F}, y) \in \mathcal{T}} \mathbf{1}(y\mathbf{W} \cdot \Phi < 1) y \Phi. \quad (11)$$

Here $\mathbf{1}(\cdot)$ is an indicator function.

C. Learning of Θ

Input to the Siamese CNN is a pair of regional features denoted as $\mathbf{f}_{i,j}$ and $\mathbf{f}_{l,k}$ along with their latent variables $v_{i,j}$ and $v_{l,k}$. We follow the method of [17] and name $(\mathbf{f}_{i,j}, \mathbf{f}_{l,k})$ a genuine pair if $v_{i,j}v_{l,k} = 1$ or an impostor pair if $v_{i,j}v_{l,k} = 0$. Obviously, concurrency values of genuine pairs should be higher than those of impostor ones. The objective function to learn the Siamese CNN is

$$\begin{aligned} L(\Theta; \mathbf{f}_{i,j}, v_{i,j}, \mathbf{f}_{l,k}, v_{l,k}) &= \frac{v_{i,j}v_{l,k}}{2} \mathbf{s}^2 + \frac{(1 - v_{i,j}v_{l,k})}{2} (\max(0, \eta - \mathbf{s}))^2, \\ s.t. \quad \mathbf{s} &= \|\mathbf{f}_{i,j} - \mathbf{f}_{l,k}\|_2, \quad v_{i,j}, v_{l,k} \in \mathbf{V}, \quad j < k, \end{aligned} \quad (12)$$

where η is a margin parameter set to 2 empirically. We adopt the standard stochastic gradient descent [33] to update

Algorithm 1 Model Learning**Input:** training samples \mathcal{T} ;**Initialization:** set learning rate $\kappa = \frac{1}{|\mathcal{T}|}$, $\mathbf{W} = [\mathbf{W}^s, \mathbf{W}^c]$;**repeat** $\forall (\mathbf{F}, y) \in \mathcal{T}$, infer \mathbf{V} using Eq. (9);compute gradient $\frac{\partial L(\mathbf{W}; \mathcal{T}, \mathbf{V})}{\partial \mathbf{W}}$ with Eq. (11), and update $\mathbf{W} = \mathbf{W} - \kappa \frac{\partial L(\mathbf{W}; \mathcal{T}, \mathbf{V})}{\partial \mathbf{W}}$;**repeat**randomly select $(\mathbf{F}, y) \in \mathcal{T}$, $\mathbf{f}_{i,j}, \mathbf{f}_{l,k} \in \mathbf{F}$ and $v_{i,j}, v_{l,k} \in \mathbf{V}$;update Θ with loss function Eq. (12);**until** Θ converges;**until** \mathbf{W} converges;**Output:** \mathbf{W}^s , \mathbf{W}^c and Θ .

the model parameter Θ and minimize Eq. (12) in the back-propagation phase.

Finally, estimation of parameters \mathbf{W}^s , \mathbf{W}^c and Θ is formulated as alternating optimization. We note this alternating optimization follows the convention of coordinate descent [34] that is easy to implement. The alternating optimization empirically converges in 3 rounds. The overall optimization is sketched in Algorithm 1.

VI. SYSTEM APPLICATION

With the model parameters, i.e., \mathbf{W}^s , \mathbf{W}^c and Θ , determined during learning, we apply the RSCM to different scenarios. In this section, we discuss the application of binary/multi-class weather classification and attribute recognition.

Inference of weather labels regarding Eq. (8) is to obtain the latent variable \mathbf{V} and label y . We use the ICM algorithm described in Section V-A to solve Eq. (8) for the latent variable. The solution is denoted as \mathbf{V}^* . Then we calculate the decision score for classification as $J(\mathbf{W}^s, \mathbf{W}^c; \mathbf{V}^*, \mathbf{F})$. When dealing with binary classification, we identify the input image as $y = 1$ if the decision score $J(\mathbf{W}^s, \mathbf{W}^c; \mathbf{V}^*, \mathbf{F}) > 0$, otherwise $y = -1$.

When extending this framework to multi-class weather classification, we apply one-vs-all strategy to train distinct decision functions of RSCM for all weather categories. For instance, when training the RSCM for the category “sunny”, we set the samples from this category as $y = 1$ and others as $y = -1$. Totally we have 6 RSCMs for the weather categories defined in our dataset. The decision scores computed by independently trained RSCMs are not in the same scale, which may bias the eventual prediction. Thus we use scaling [35] to transform the decision score into a probability. When testing a new image, we assign the input image to the weather category with the highest probability of $y = 1$.

We note our framework can be adapted to weather attribute recognition. Similar to multi-class classification, we again use one-vs-all strategy to train a separate RSCM for each attribute. For a certain attribute, we set the training images with this attribute as $y = 1$ and others as $y = -1$. Repeating this process, we have RSCMs for all the attributes. Scaling [35] is also used to compute the probability that predicts an image to have the corresponding attribute. When testing a new image,

we set a threshold probability. Any attribute with higher probability than the threshold is regarded as “existence”.

VII. EXPERIMENTS

In this section, we evaluate proposed method on our multi-class weather and two-class weather [3] datasets with comprehensive analysis. We begin with some experimental settings.

Since our RSCM depends in part on the regions partitioned by the co-segmentation tool [26], we randomly select 2,000 images from the training set regardless of weather categories. We use K-means to partition these 2,000 images based on HoG [29] and GIST features [10]. 10 groups of images are produced. The images in each group are used to train a co-segmentation model.

VGG-16 [36] pre-trained on ImageNet classification [33] serves as the CNN architecture in our RSCM. It is built on Caffe platform [37]. Without pre-training, the network yields 9.1% and 15.8% performance drop on weather classification and attribute recognition tasks respectively. To extract regional feature, we first perform CFM [30] on the last convolutional layer of CNN. Then we use a 7×7 spatial pooling to obtain the fixed-length feature. This feature passes two fully connected layers to form the eventual 4,096D regional feature.

A. Experiments on Our Weather Classification Dataset

We evaluate our RSCM on our multi-class weather dataset. We provide 10 training/testing splits. In each split, we randomly select 50% of the data from each weather category for training and the rest for testing. These splits are used in all our experiments. Following the evaluation metric of [3], we report the mean and variance of the normalized classification accuracy.

1) *Sensitivities to Region Generation:* Generation of image regions with the co-segmentation tool [26] is subject to the number of region clusters. We investigate how sensitive our method is in the process of region generation, regarding different settings of region clusters. When evaluating the performance of RSCM, we diversify the number of the region clusters $M \in \{3, 6, 9, 12, 15\}$, and report the classification accuracy for every case. In each case, we show the average inference time per image.

In each training split, we further select 50% of the images for validation. Table I shows the results of our method using

TABLE I

SENSITIVITIES TO THE NUMBER OF REGION CLUSTERS EVALUATED ON OUR VALIDATION SETS. “# REGION CLUSTERS” SHOWS THE NUMBER OF REGION CLUSTERS WHEN WE USE CO-SEGMENTATION. “AVG. INFERENCE TIME” SHOWS THE AVERAGE INFERENCE TIME PER IMAGE

# region clusters	avg. inference time (ms)	accuracy (%)
$M = 3$	120	82.5 ± 1.1
$M = 6$	356	83.3 ± 1.0
$M = 9$	1,217	85.1 ± 0.7
$M = 12$	3,866	90.7 ± 1.4
$M = 15$	8,442	88.3 ± 2.2

TABLE II

CLASSIFICATION ACCURACY (%) OF DIFFERENT COMBINATIONS OF COMPONENTS ON OUR TEST SETS

method	selection	concurrency	RSCM
accuracy (%)	87.6 ± 0.7	50.3 ± 1.1	94.1 ± 0.3

different numbers of region clusters on the validation sets. We note that the time cost is increasing as the number of region clusters becomes larger. This is because the algorithm needs more iterations to converge with more region clusters.

We empirically observe that our RSCM performs the best when $M = 12$. More region clusters, e.g. $M = 15$, reduce the overall classification accuracy. It may be because too many region clusters produce a lot of tiny regions without useful information. We thus keep $M = 12$ in our experiments.

2) *Component Analysis*: Our joint model consists of selection and concurrency components. We show they are both necessary. When removing one of them from the whole framework, the overall performance drops.

The first experiment is to apply the full version of RSCM. The classification accuracy is listed in the column *RSCM* of Table II. The accuracy is 94.1 ± 0.3 .

Then we only incorporate region selection in Eq. (8) in our RSCM. This case corresponds to removing region concurrency in the inference and learning phases. We list the classification results in the column *selection* of Table II. Compared to the complete framework, training RSCM only with selection component shows inferior classification accuracy. This is because the update of the CNN is not aware of the regions with high concurrency. As such, the regional feature extracted with the CNN provides inaccurate concurrency information. It leads to degradation of classification performance.

Finally, we remove region selection in Eq. (8). The accuracies are listed in the column *concurrency* of Table II. Though the CNN is updated with region concurrency in the training phase, we observe significant performance drop compared to RSCM. This is because the region concurrency to optimize CNN is not discriminative enough, without the guidance of region selection.

These experiments manifest the necessity and usefulness of both the region selection and concurrency components in terms of the classification performance.

3) *Strategies of Utilizing Regions*: In Table III, we compare our method with other strategies of utilizing regions.

TABLE III

CLASSIFICATION ACCURACY ON OUR TEST SETS VIA DIFFERENT STRATEGIES OF UTILIZING REGIONS. “MAX” AND “AVG” MEAN MAX AND AVERAGE POOLING. “W/O REGION CLUSTER” DOES NOT CONSIDER REGION CLUSTERS WHILE “W/ REGION CLUSTER” DOES

method	strategy	accuracy (%)
whole image	global	91.5 ± 0.2
w/o region cluster (MAX)	local	70.9 ± 0.2
w/o region cluster (AVG)	local	73.1 ± 0.6
w/ region cluster (MAX)	local	82.6 ± 0.7
w/ region cluster (AVG)	local	76.3 ± 0.2
RSCM	local	94.1 ± 0.3

TABLE IV

COMPARISON WITH OTHER METHODS ON OUR TEST SETS

method	accuracy (%)
Roser et al. [2]	13.7 ± 1.5
Yang et al. [39]	22.7 ± 0.6
Wang et al. [40]	28.3 ± 1.1
Krizhevsky et al. [33]	80.3 ± 1.0
Simonyan et al. [36]	91.5 ± 0.2
RSCM	94.1 ± 0.3

Our method considers different region clusters. It jointly selects the discriminative regions with high concurrency and update the model parameters. To show the effectiveness of exploiting regions, we use the whole image to fine-tune the VGG-16 and evaluate the performance on our dataset as a baseline. This is a global strategy because the whole image, rather than some of the local regions, is used to form the final feature. The accuracy of this baseline is 91.5 ± 0.2 . The comparison with our method shows that utilizing local regions is vital to weather classification.

There are two simpler alternatives to utilize local regions. Initially, we follow the steps of CFM [30] to fine-tune the classification network using the local regions. We use the network to extract regional feature. The first alternative is to use common max and average pooling methods to combine all regional features regardless of their region clusters. Both pooling methods generate 4,096D features given an image. The second alternative considers region clusters. Pooling is applied to combine regional features of the same class. Then they are concatenated as a 49,116D ($4,096 \times 12$) feature. The features generated using the above two alternatives are used to train a multi-class SVM [38]. These two local strategies, as shown in Table III, are not optimal for weather classification. It is because merging all regions rather than selecting useful ones for classification loses information.

4) *Comparisons With Other Methods*: In Table IV, we compare with other widely-used classification methods [2], [33], [36], [39], [40]. We note that the accuracies yielded by the non-CNN based methods [2], [39], [40] are left far behind the CNN-based ones [33], [36]. This is because CNN fits large image data [41] and transfers the pre-trained knowledge to the weather classification task. Our method takes the advantage of the pre-trained CNN model. It yields accuracy 94.1 ± 0.3 .

TABLE V
COMPARISON WITH OTHER METHODS ON OUR ATTRIBUTE DATASET

	sunny	cloudy	rainy	snowy	haze	thunder	average accuracy
Roser et al. [2]	12.3 ± 0.5	11.2 ± 0.6	35.7 ± 1.5	3.7 ± 1.2	5.6 ± 1.3	1.8 ± 0.3	10.5 ± 1.7
Wang et al. [40]	38.3 ± 1.8	14.7 ± 0.9	22.5 ± 1.8	12.4 ± 0.7	23.6 ± 3.1	11.7 ± 1.6	20.8 ± 2.2
Yang et al. [39]	30.5 ± 3.2	23.8 ± 2.2	30.6 ± 1.3	14.7 ± 1.5	21.6 ± 1.2	14.5 ± 2.0	22.8 ± 2.6
Krizhevsky et al. [33]	78.6 ± 0.9	62.8 ± 1.2	59.2 ± 2.3	70.6 ± 1.2	60.7 ± 1.3	69.6 ± 2.2	67.2 ± 1.8
Simonyan et al. [36]	82.5 ± 0.6	73.6 ± 0.8	66.1 ± 3.6	73.0 ± 0.5	65.8 ± 2.4	76.1 ± 2.1	73.0 ± 2.4
RSCM	90.7 ± 1.3	86.2 ± 1.6	84.6 ± 2.7	89.3 ± 1.1	84.7 ± 2.2	75.5 ± 3.2	85.2 ± 1.5

B. Experiments on Our Weather Attribute Dataset

In this part, our RSCM is evaluated on our weather attribute dataset that contains 5,000 images. We randomly select 2,500 images for training and the rest for testing. This process is repeated for 10 times. We follow the previous attribute recognition tasks [42]–[44] to report the normalized per-attribute accuracy and their variance. The average accuracy on all attributes is also reported.

We note the baseline VGG-16 model can be trivially apply to weather attribute recognition. With this setting, the sigmoid cross entropy loss is used as the objective to allow the multiple attributes during training. The VGG-16 model outputs the independent probability for each attribute. If a probability exceeds a threshold, it is predicted as being present. In Figure 14, we compare our RSCM with the baseline VGG-16 model. We report the average recognition accuracy under different thresholds. As the threshold increase, we observe the performance of all methods is improved. This is because more false positive predictions are rejected by the higher threshold. When the threshold is too high, some true positive predictions are missed. It reduces performance. In this comparison, RSCM performs better than the baseline VGG-16 model in most cases.

In Table V, we compare our RSCM with other methods. The features of [2], [39], and [40] are used to train separate linear SVM [38] for each attribute. The scores output by SVM are all transformed to probabilities [35]. We also report the performance of two CNN-based methods [33], [36], which perform attribute recognition on the whole image. Our RSCM outperforms other methods. It suggests that the discriminative regions proposed by our RSCM are vital for the difficult tasks like weather attribute recognition.

C. Study on Weather Classification and Attribute Recognition

In the above experiments, our RSCM is applied to weather classification and attribute recognition tasks. In what follows, we further study our model in these two tasks where RSCM can effectively mine useful information from one task to help the other.

In classification, our RSCM is trained for each weather category. Although our RSCM is originally applied to predict a single label for each image, it can naturally provide confidence for all weather conditions. We apply our RSCM trained on classification set to test on the attribute set. We report the attribute recognition results using the classification RSCM

TABLE VI
RESULTS OF CLASSIFICATION, ATTRIBUTE AND AUGMENTED RSCM ON THE ATTRIBUTE DATASET

method	accuracy (%)
classification RSCM	70.3 ± 1.7
attribute RSCM	85.2 ± 1.5
augmented RSCM	88.6 ± 1.1

TABLE VII
RESULTS OF CLASSIFICATION, ATTRIBUTE AND AUGMENTED RSCM ON THE CLASSIFICATION DATASET

method	accuracy (%)
attribute RSCM	88.5 ± 1.6
classification RSCM	94.1 ± 0.3
augmented RSCM	96.4 ± 0.7

in Table VI. They are with lower accuracy than what the attribute RSCM produces. Yet we observe the classification RSCM has relatively high response to existing attributes. A few examples are shown in Figure 13 with their attribute confidence. This observation manifests that the classification set contains useful information. It can be mined to help attribution recognition. When using both the classification and attribute sets to train RSCM, the results (listed in row *augmented RSCM* of Table VI) are better.

We also apply the attribute RSCM to the classification set. In this scenario, we select the most confident attribute as the weather class. The classification accuracy of the attribute RSCM is listed in Table VII. The result of the attribute RSCM is already similar to that of classification RSCM, by considering the fact that the training attribute set (2,500 images) is smaller than the classification set (30,000 images). It suggests that the RSCM learning of weather correlation benefits classification. When combining the classification and attribute sets for training, our augmented RSCM yields better accuracy as reported in Table VII. We believe more attribute images can further strengthen the ability.

D. Experiments on Two-Class Weather Dataset

We evaluate our method on the other weather dataset [3]. This dataset contains 10,000 images divided into two categories, i.e. sunny and cloudy. We follow the training/testing scheme of [3] where 80% of the data (4,000 sunny and 4,000 cloudy images) are randomly selected for training

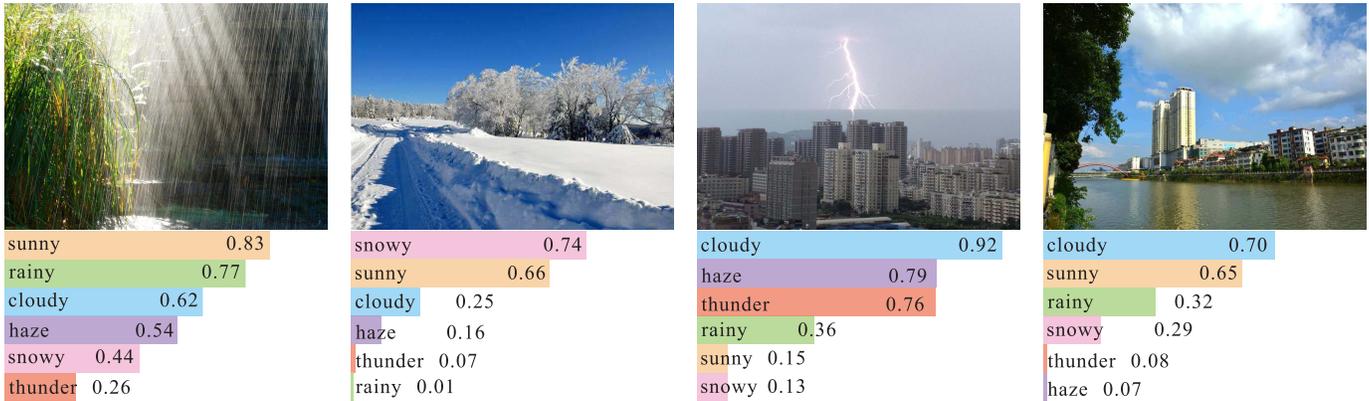


Fig. 13. Sample images with attribute confidence output from our classification RSCM. We sort the confidence in a descending order.

TABLE VIII
COMPARISON WITH STATE-OF-THE-ARTS ON
TWO-CLASS WEATHER DATASET

Method	accuracy (%)
Yan et al. [1]	24.6 ± 2.6
Roser et al. [2]	26.2 ± 2.3
Lalonde et al. [19]	39.5 ± 2.3
Lu et al. [3]	53.1 ± 2.2
Elhoseiny et al. [45]	91.1 ± 1.8
Simonyan et al. [36]	97.2 ± 0.6
RSCM	99.6 ± 0.4

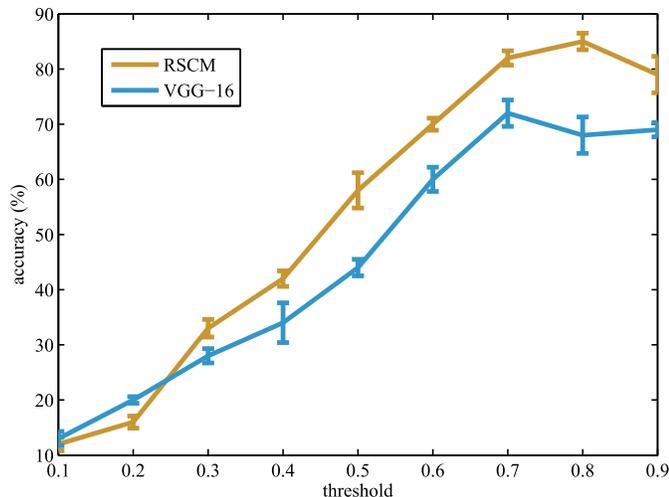


Fig. 14. Average accuracies of RSCM and VGG-16 models on our attribute dataset under different thresholds.

and the rest are for testing in each iteration. We repeat training/testing for 5 iterations. In addition, we supply the classification results when using VGG-16. The VGG-16 result on this dataset is 97.2 ± 0.6 . Our method yields 99.6 ± 0.4 .

VIII. CONCLUSION AND FUTURE WORK

We have presented a new region selection and concurrency model for weather classification and attribute recognition. It selects discriminative regional features from different

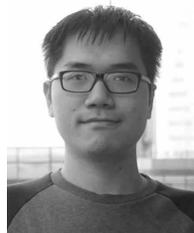
region clusters, while effectively discovering their feature concurrencies that are common for outdoor images to understand weather conditions. We have also contributed a novel multi-class dataset to facilitate study on weather recognition.

Region selection and concurrency model is the core idea of this paper. Currently, our model discovers regions from all region clusters. However, there may exist some clusters that have little or no discriminative regions. Thus using all region clusters requires unnecessary overhead to recognize an image, and some useless regions may even harm the performance. It makes our method less practical to real-time applications, e.g., video-based weather recognition. In future work, we will make generation of region clusters aware of the weather condition during training. It allows the model to learn the pattern of useful region clusters. This idea can hopefully reduce the problematic clusters and increase accuracy of weather recognition.

REFERENCES

- [1] X. Yan, Y. Luo, and X. Zheng, "Weather recognition based on images captured by vision system in vehicle," in *Advances in Neural Networks—ISNN* (Lecture Notes in Computer Science), vol. 5553. Wuhan, China: Springer, 2009, pp. 390–398.
- [2] M. Roser and F. Moosmann, "Classification of weather situations on single color images," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2008, pp. 798–803.
- [3] C. Lu, D. Lin, J. Jia, and C.-K. Tang, "Two-class weather classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 3718–3725.
- [4] Z. Zhang and H. Ma, "Multi-class weather classification on single images," in *Proc. IEEE ICIP*, Sep. 2015, pp. 4396–4400.
- [5] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *Proc. IEEE CVPR*, Jun. 2010, pp. 3485–3492.
- [6] N. Zhang, J. Donahue, R. B. Girshick, and T. Darrell, "Part-based R-CNNs for fine-grained category detection," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), vol. 8689. Zurich, Switzerland: Springer, 2014, pp. 834–849.
- [7] D. Lin, C. Lu, R. Liao, and J. Jia, "Learning important spatial pooling regions for scene classification," in *Proc. IEEE CVPR*, Jun. 2014, pp. 3726–3733.
- [8] D. Lin, X. Shen, C. Lu, and J. Jia, "Deep LAC: Deep localization, alignment and classification for fine-grained recognition," in *Proc. IEEE CVPR*, Jun. 2015, pp. 1666–1674.
- [9] H. Katsura, J. Miura, M. Hild, and Y. Shirai, "A view-based outdoor navigation using object recognition robust to changes of weather and seasons," in *Proc. IEEE IROS*, Oct. 2003, pp. 2974–2979.

- [10] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.
- [11] J. Xiao *et al.*, "Basic level scene understanding: Categories, attributes and structures," *Frontiers Psychol.*, vol. 4, p. 506, Aug. 2013.
- [12] P. Isola and C. Liu, "Scene collaging: Analysis and synthesis of natural images with semantic layers," in *Proc. IEEE ICCV*, Dec. 2013, pp. 3048–3055.
- [13] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE CVPR*, Jun. 2006, pp. 2169–2178.
- [14] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *Computer Vision—ECCV (Lecture Notes in Computer Science)*, vol. 8695. Zurich, Switzerland: Springer, 2014, pp. 392–407.
- [15] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
- [16] S. N. Parizi, J. G. Oberlin, and P. F. Felzenszwalb, "Reconfigurable models for scene recognition," in *Proc. IEEE CVPR*, Jun. 2012, pp. 2775–2782.
- [17] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE CVPR*, Jun. 2005, pp. 539–546.
- [18] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE CVPR*, Jun. 2006, pp. 1735–1742.
- [19] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan, "Estimating the natural illumination conditions from a single outdoor image," *Int. J. Comput. Vis.*, vol. 98, no. 2, pp. 123–145, 2012.
- [20] S. J. Kim, J. Frahm, and M. Pollefeys, "Radiometric calibration with illumination change for outdoor scene analysis," in *Proc. IEEE CVPR*, Jun. 2008, pp. 1–8.
- [21] S. Wehrwein, K. Bala, and N. Snavely, "Shadow detection and sun direction in photo collections," in *Proc. IEEE 3DV*, Oct. 2015, pp. 460–468.
- [22] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Scene parsing with multiscale feature learning, purity trees, and optimal covers," in *Proc. ICML*, 2012, pp. 1–8.
- [23] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *Proc. IEEE ICCV*, Sep. 2009, pp. 1–8.
- [24] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2368–2382, Dec. 2011.
- [25] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "ScribbleSup: Scribble-supervised convolutional networks for semantic segmentation," in *Proc. IEEE CVPR*, Jun. 2016, pp. 3159–3167.
- [26] A. Joulin, F. R. Bach, and J. Ponce, "Multi-class cosegmentation," in *Proc. IEEE CVPR*, Jun. 2012, pp. 542–549.
- [27] J. C. Rubio, J. Serrat, A. M. López, and N. Paragios, "Unsupervised cosegmentation through region matching," in *Proc. IEEE CVPR*, Jun. 2012, pp. 749–756.
- [28] A. Joulin, F. R. Bach, and J. Ponce, "Discriminative clustering for image co-segmentation," in *Proc. IEEE CVPR*, 2010, pp. 1943–1950.
- [29] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE CVPR*, Jun. 2005, pp. 886–893.
- [30] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proc. IEEE CVPR*, Jun. 2016, pp. 3150–3158.
- [31] R. B. Girshick, "Fast R-CNN," in *Proc. IEEE ICCV*, Jun. 2015, pp. 1440–1448.
- [32] J. Besag, "On the statistical analysis of dirty pictures," *J. Roy. Statist. Soc. Ser. B, Methodol.*, vol. 48, no. 3, pp. 259–302, 1986.
- [33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.
- [34] S. J. Wright, "Coordinate descent algorithms," *Math. Programm.*, vol. 151, no. 1, pp. 3–34, 2015.
- [35] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Adv. Large Margin Classifiers*, vol. 10, no. 3, pp. 61–74, 1999.
- [36] K. Simonyan and A. Zisserman. (Sep. 2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [37] Y. Jia *et al.* (Jun. 2014). "Caffe: Convolutional architecture for fast feature embedding." [Online]. Available: <https://arxiv.org/abs/1408.5093>
- [38] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.
- [39] J. Yang, K. Yu, Y. Gong, and T. S. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE CVPR*, Jun. 2009, pp. 1794–1801.
- [40] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE CVPR*, Jun. 2010, pp. 3360–3367.
- [41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE CVPR*, Jun. 2009, pp. 248–255.
- [42] T. Berg and P. N. Belhumeur, "POOF: Part-based One-vs.-One features for fine-grained categorization, face verification, and attribute estimation," in *Proc. IEEE CVPR*, Jun. 2013, pp. 955–962.
- [43] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE ICCV*, Jun. 2015, pp. 3730–3738.
- [44] A. Yu and K. Grauman, "Just noticeable differences in visual attributes," in *Proc. IEEE ICCV*, Jun. 2015, pp. 2416–2424.
- [45] M. Elhoseiny, S. Huang, and A. M. Elgammal, "Weather classification with deep convolutional neural networks," in *Proc. IEEE ICIP*, Sep. 2015, pp. 3349–3353.



Di Lin received the bachelor's degree in software engineering from Sun Yat-sen University in 2012 and the Ph.D. degree from The Chinese University of Hong Kong in 2016. He is currently an Assistant Professor with the College of Computer Science and Software Engineering, Shenzhen University. His research interests are computer vision and machine learning.



Cewu Lu received the B.S. and M.S. degrees from the Chongqing University of Posts and Telecommunications and the Graduate University of Chinese Academy of Sciences in 2006 and 2009, respectively, and the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong in 2013. He was a Research Fellow with The Hong Kong University of Science and Technology. He was a Research Fellow with the Stanford University AI Laboratory. Since 2016, he has been with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, where he is currently a Professor. His current research interests include activity recognition, object detection, and image/video processing. His team received the fourth place in ILSVRC 2014 among the 38 participating teams. He received the Best Paper Award of NPAR 2012 and served as an Associate Editor of the journal *gtCVPR* and reviewers for several major computer vision and graphics conferences and journals, such as *TPAMI* and *TOG*.



Hui Huang received the Ph.D. degree in computational math from Wuhan University in 2006 and the Ph.D. degree in applied math from The University of British Columbia in 2008. She is currently a Distinguished Professor with Shenzhen University, where she also directs the Visual Computing Research Center with the College of Computer Science and Software Engineering. She is currently a Senior Member of ACM, a Distinguished Member of CCF, and an Associate Editor-in-Chief of *The Visual Computer*. She was a recipient of NSFC Excellent Young

Researcher Program and Guangdong Technology Innovation Leading Talent award in 2015. She is on the Editorial Board of *Computer Graphics Forum* and *Frontiers of Computer Science*.



Jiaya Jia received the Ph.D. degree in computer science from The Hong Kong University of Science and Technology in 2004. He is currently a Professor with the Department of Computer Science and Engineering, The Chinese University of Hong Kong (CUHK). He is also the Head of the Research Group, focusing on computational imaging, machine learning, practical optimization, and low and high-level computer vision. He received the Young Researcher Award 2008 and the Research Excellence Award 2009 from CUHK. He served as the Area Chair

for several-year ICCV and CVPR. He was also on the technical paper program committees of SIGGRAPH, SIGGRAPH Asia, ICCP, and 3DV, and co-chaired the Workshop on Interactive Computer Vision, in conjunction with ICCV 2007. He currently serves as an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.