Relative CNN-RNN: Learning Relative Atmospheric Visibility From Images

Yang You, Cewu Lu, *Member, IEEE*, Weiming Wang, and Chi-Keung Tang, *Senior Member, IEEE*

***Abstract*— We propose a deep learning approach for directly estimating relative atmospheric visibility from outdoor photos without relying on weather images or data that require expensive sensing or custom capture. Our data-driven approach capitalizes on a large collection of Internet images to learn rich scene and visibility varieties. The relative CNN–RNN coarse-to-fine model, where CNN stands for convolutional neural network and RNN stands for recurrent neural network, exploits the joint power of relative support vector machine, which has a good ranking representation, and the data-driven deep learning features derived from our novel CNN–RNN model. The CNN–RNN model makes use of shortcut connections to bridge a CNN module and an RNN coarse-to-fine module. The CNN captures the global view while the RNN simulates human's attention shift, namely, from the whole image (global) to the farthest discerned region (local). The learned relative model can be adapted to predict absolute visibility in limited scenarios. Extensive experiments and comparisons are performed to verify our method. We have built an annotated dataset consisting of about 40 000 images with 0.2 million human annotations. The large-scale, annotated visibility data set will be made available to accompany this paper.**

***Index Terms*— Convolutional neural network, recurrent neural network, deep learning, atmospheric visibility, relative attributes learning, large-scale image collection.**

I. Introduction

SMOG pollution has become a global health and environment concern. For example, Indonesian forest fires have posed recurring air pollution problems in Singapore and Malaysia. Volcanic eruptions in Iceland in 2010 had caused enormous disruption to air traffic across Europe. Atmospheric visibility may change drastically in a matter of minutes which calls for real-time visibility monitors in air traffic control, pollution monitoring, and accident detection (e.g., fire accident or arson).

Atmospheric visibility is measured by weather observatories. However, observatories are geographically sparse and their reported visibilities are typically in hours of delay. Today, surveillance cameras are abound, and thousands of geo-tagged outdoor images are uploaded to social media in each second. Estimating atmospheric visibility from a photo has a high potential in real-time and ubiquitous monitoring of smog and air pollution. Different from accurate measurements obtained from expensive equipment in weather stations, which are used for accurate scientific calculations and analysis, this affordable, image-based visibility estimation serves a different and important application, namely, timely monitoring of atmospheric visibility conditions.

(a) (b)

Fig. 1. An image pair for visibility comparison, where (a) clearly has a better visibility than (b); it is hard for humans to specify absolute visibility/depth from single images.

A. Relative Visibility

In this paper we propose to estimate *relative* visibility from single photos, where zero relative visibility indicates absolute invisibility and one indicates clearly visible scene. As we shall explain, our learned relative model can be adapted to estimate absolute visibility in a limited and small training data scenario. We believe relative visibility is as useful as relative humidity which we are used to. Ideally, absolute visibility should be estimated, which is the farthest distance at which the pertinent scene is still discernable. However, it is well known that human specification of absolute depth from single images is very inaccurate [1]. Another problem is the lack of meteorological images (or simply images) with sufficient scene variety and absolute visibility measurement, due to the limited number of weather observatories and stations (or landmarks with known distance from the capture camera) around the world. A general prediction model cannot be reliably learned from sparse training data.

On the other hand, for humans it is easy to accurately label relative attributes. For example, in Figure 1, without

Manuscript received September 8, 2016; revised July 28, 2017; accepted October 24, 2017. Date of publication July 18, 2018; date of current version September 19, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Gustavo K. Rohde. *(Corresponding author: Weiming Wang.)*

Y. You and W. Wang with the Department of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: wangweiming@sjtu.edu.cn).

C. Lu is with the Department of Computer Science, Shanghai Jiao Tong University, Shanghai 200240, China.

C.-K. Tang is with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, 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.2018.2857219

Fig. 2. The coarse-to-fine sequence. The attention region shrinks from the whole image (1) to the farthest discerned region (7).

any distance measurement (a) clearly has a better visibility than (b). Thus, to prepare the training data, we propose to annotate image pairs by ranking their visibility directly observed from the images. This makes it possible for accurately labeling a large volume of image data for training an accurate ranking model. Another advantage is that we can substantially increase the number of ordering constraints, by applying an image dehazing filter to automatically generate new pairwise constraints, when the training pair is labeled as "unordered" by a human annotator who either sees no difference or cannot rank the visibility in the given pair.

B. Relative CNN-RNN

We propose the relative CNN-RNN model, where CNN stands for convolutional neural network and RNN stands for recurrent neural network, and show that it significantly outperforms the typically used ranking support vector machine (SVM), existing machine-learned ranking algorithms, and state-of-the-art image processing techniques without scene semantics consideration and/or using large-scale training data.

Our relative CNN-RNN coarse-to-fine attention model attempts to simulate human's visual perception on atmospheric visibility, in both coarse and fine levels. Echoing how humans can obtain a coarse estimation of the haze density by the global appearance of the scene, the CNN architecture is responsible for learning the overall visibility from the whole image. With a global view, we then start to look for the farthest discerned region (or object) in the image to determine the scene visibility where visual processing occurs in a finer level. This is in fact coherent with the definition of atmospheric visibility used in meteorology: visibility measures the farthest distance at which an object or light can still be discerned. Inspired by [2], we model this coarse-to-fine process by RNN to simulate the pertinent coarse-to-fine visual attention shift exhibited by humans. The coarse-to-fine attention transition (an example is illustrated in Figure 2) can incorporate richer and more detailed visibility information in comparison to the use of a global representation such as the CNN feature alone.

It turns out the two visibility description models, namely, the CNN which represents coarse visibility description, and RNN which represents the coarse-to-fine attention shift, can be coherently integrated. The RNN model starts from a global region gradually shrinking to a local region, while the CNN model reversely starts from local description to finally obtain a global representation. Though their goals are different, the two models share local representations in different spatial regions. In our framework, we allow the information to flow among the two models, that is, the two models "speak" with each other to collaboratively achieve visibility recognition.

C. Results

We have conducted extensive experiments on both large-sized relative dateset and small-sized absolute dataset as well. Our proposed model achieves good results in atmospheric visibility estimation when compared to a set of baseline solutions, existing machine-learned ranking algorithms and state-of-the-art image processing techniques. We will release our annotated visibility dataset for benchmarking, which consists of about 40,000 images and 0.2 million human annotations. As new images can be added easily this dataset is readily scalable.

II. Related Work

A. Atmospheric Visibility Estimation

The amount of existing work is small on leveraging images to automatically estimate atmospheric visibility. Moreover, the previously proposed methods mainly relied on low-level image cues (e.g., image gradients, contrast, hue, saturation, etc) without adequate scene consideration or understanding. **Though recent single image depth estimation is improved by deep learning, e.g., in [3] depth is regressed from patch patterns, the accuracy of the estimated depth still falls short for visibility estimation.** They also required various parameter settings or manual specification of visual targets as reviewed in [4]. Baumer *et al*. [5] estimated visibility by measuring the loss of edges of pre-selected known objects in panoramic images. In [6] a probabilistic based approach was presented that takes into account the distribution of contrast in the scene where the Lambertian scene assumption was used. **We prefer a learning-based approach which can be generally applied or adopted to different scenes**. For example, given the same haze density, a photo depicting a downtown scene with skyscrapers look quite different from one depicting mountains and ocean in an open space.

B. Image Dehazing

On the other hand there is a sizable amount of work on image dehazing. No existing methods however have capitalized on large image collection to learn scene visibility from a great variety of examplars to make the solution more robust against scene variance. The geometry-based approach [7]–[9] requires 3D or depth which is acquired from range sensors or rough estimation is made by the user. In [10], [11], the haze removal processing leverages polarized filters applied during multiple capture of the same scene. In [12], multiple

images of the same scene were analyzed which were shot under different weather conditions. Recent advances in single-image dehazing used the powerful dark channel prior [13], powerful optimization [14], [15], practical assumptions on local contrast [16] and albedos [17], new image filters such as the atmospheric point spread function or filters derived from generalized Gaussian distribution [18], [19], a panorama alongside with user annotation and calibration [5], probability distribution model on image contrast [6], and a log-linear model relating transmission and extinction of light [4]. A variety of image-based haze features were investigated in [20] using regression in a random forest framework. We will show in the experimental section that although single image dehazing may seem to be a plausible approach, it cannot be used for accurate estimation of atmospheric visibility.

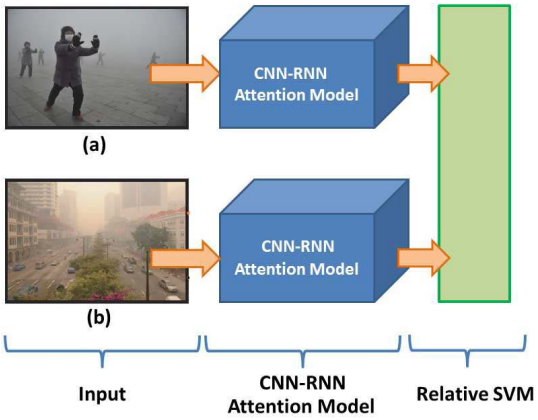

Fig. 3. Our relative CNN-RNN architecture for an image pair; (a) and (b) are input image pairs.

III. Data Collection and Annotation

A. Relative Visibility Dataset

Our goal is to build a dataset covering a large number and variety of scenes; this dataset should be readily scalable to grow and expand with more images. Our first attempt consisted of collecting images with visibility measurement obtained from weather observatories and stations around the world. But observatories are sparsely located and the variety of scenes is very limited. Furthermore, relying on photos taken at observatories is not scalable as we believe not many observatories/weather stations will be built in the coming years. While other sensors such as visibility meters can be used, since our goal is to estimate visibility from a single image, we cannot take into account these sensor signals in the testing phase.

We next turned to collect outdoor images from the internet and annotate them by people with environment science training. We used the keyword "fog", "haze", "mist" and "smog" in Flickr and downloaded about 60,000 images. We discarded low-quality images and included clear and haze-free images which are selected manually. Finally, we collected 37,420 images with different visibility levels ranging from clearly visible to heavily smoggy.

To annotate the training images, our first attempt was to ask annotators to estimate the "farthest distance at which an object or light can be discerned" given a single image. But we found that the annotation varies significantly from person to person. For example, the human visibility annotation of Figure 1(a) ranged from 300, 400, 600 to 800 meters. This stems from the fact that human vision in general cannot give accurate absolute depth measurement, much less that it is difficult to estimate depth from a 2D image.

It is however easier for us to deduce correct relative relationship without ambiguity, or in our case rank two given images based on the observed visibility. For example, Figure 1(a) has a better visibility than (b) without absolute visibility estimation. Thus, image pairs annotation was adopted, that is, we asked annotators to rank a given image pair in visibility. In the ambiguous case where the user cannot decide, the image pair will be labeled as "similar visibility." A total of 7 subjects was asked to produce 224,520 image pairs annotations. We then substantially increase the number of ordering constraints automatically by employing a dehaze filter which will be described shortly.

We also ask the subjects to annotate the farthest discerned regions of the images. This information will be used in our coarse-to-fine model training.

B. Absolute Visibility Dataset

We have prepared a small dataset of absolute visibility measurement captured at a small number of (pixel) locations, courtesy of the Hong Kong Observatory. Not surprisingly, their cameras were fixed and captured a very limited number of scenes. While their cameras captured images at hourly intervals, most of the images are highly redundant since each camera captured the same scene over each passing year. While approximately 500,000 images with absolute visibility readouts were collected, after removing highly redundant images only 3,146 images remain to form the small absolute visibility dataset. The unit of visibility measurement is metres.

IV. Relative CNN-RNN Coarse-to-Fine Model

We propose the **relative CNN-RNN model** to solve our machine-learned ranking problem. In this section, we first revisit ranking SVM and introduce how to enrich the training data by applying a dehazing filter on unordered pairs. We then describe our novel CNN-RNN architecture. Finally, we present how to embed this architecture in ranking SVM. Figure 3 gives an overview of our relative visibility learning framework.

A. Relative SVM

In relative SVM, we define a set of training images $I_i, i = 1, \ldots, n$, represented in $\mathbb{R}^m$ by feature $f(I_i)$. We are also given a set of ordered image pairs $\mathcal{O}$ and a set of unordered pairs $\mathcal{S}$, such that $(i, j) \in \mathcal{O} = i \succ j$, that is, image i has a better visibility than j, and $(i, j) \in \mathcal{U} = (i = j)$ means that i and j have similar visibility or the annotator cannot decide the order. Our goal is to learn a visibility score function

$$r(I) = \mathbf{w}^T f(I), \tag{1}$$

where $f(I)$ is a feature computed on image I and $\mathbf{w}^T$ is feature weight, such that the number of the following requirements satisfied should be maximized:

$$\forall (i,j) \in \mathcal{O}, \quad \mathbf{w}^T f(I_i) \geq \mathbf{w}^T f(I_j), \tag{2}$$

$$\forall (i,j) \in \mathcal{U}, \quad \mathbf{w}^T f(I_i) = \mathbf{w}^T f(I_j). \tag{3}$$

While this is an NP hard problem, it is possible to approximate the solution with the introduction of nonnegative slack variables as similarly done in SVM classification. We adopt the formulation in [21], which was originally applied to webpage ranking, leading to the following optimization problem:

$$\min_{\mathbf{w}} \ \|\mathbf{w}\|_2^2 + C\Big(\sum_{(i,j)\in\mathcal{O}} \xi_{i,j} + \sum_{(i,j)\in\mathcal{U}} \gamma_{i,j}\Big) \tag{4}$$

$$s.t.\ \forall (i,j) \in \mathcal{O}, \quad \mathbf{w}^T f(I_i) - \mathbf{w}^T f(I_j) \geq 1 - \xi_{i,j} \tag{5}$$

$$\forall (i,j) \in \mathcal{U}, \quad |\mathbf{w}^T f(I_i) - \mathbf{w}^T f(I_j)| = \gamma_{i,j}, \tag{6}$$

where $\xi_{i,j}$ and $\gamma_{i,j}$ are errors standard in SVM formulation and C is a constant parameter. With the learned SVM model, given an image I with feature $f(I)$ its visibility score is given by $\mathbf{w}^T f(I)$.

B. Inferring Additional Ordering Constraints

Scene visibility is a complex visual concept requiring a large number of image pair constraints in training to avoid undesirable over-fitting. We propose to use a state-of-the-art image dehazing filter to automatically generate new pairwise constraints from the given training pairs labeled as "unordered," where the human annotator cannot differentiate the relative visibility ranking. As done before, we remove the image pairs with haze-free image from the unordered image pair set $\mathcal{U}$ and denote the resulting set $\mathcal{B}$.

Denoting $\varphi[\cdot]$ as a dehazing filtering operation, we can immediately produce new order constraints

$$\forall (i,j) \in \mathcal{B}, \quad \mathbf{w}^T f(\varphi[I_i]) > \mathbf{w}^T f(I_j), \tag{7}$$

$$\forall (i,j) \in \mathcal{B}, \quad \mathbf{w}^T f(I_i) < \mathbf{w}^T f(\varphi[I_j]). \tag{8}$$

Currently state-of-the-art dehaze filters are excellent in enhancing visibility given a hazy image. The dehaze filter we use is [13] with a guided filter. Figure 4 shows an example of a new order pair produced. Though, few of faircases in dehaze, they still don't degrade the overall training preformance in the viewpoint of statistics.

In response to the newly added constraints the relative SVM can be expressed as,

$$\begin{aligned}
&\min_{\mathbf{w}} \|\mathbf{w}\|_2^2 + C\Big[\sum_{(i,j)\in\mathcal{O}} \xi_{i,j} + \sum_{(i,j)\in\mathcal{U}} \gamma_{i,j} + \sum_{(i,j)\in\mathcal{B}} (\eta^+_{i,j} + \eta^-_{i,j})\Big] \\
&s.t.\ \forall (i,j) \in \mathcal{O}, \quad \mathbf{w}^T f(I_i) - \mathbf{w}^T f(I_j) \geq 1 - \xi_{i,j}, \\
&\quad \forall (i,j) \in \mathcal{U}, \quad |\mathbf{w}^T f(I_i) - \mathbf{w}^T f(I_j)| = \gamma_{i,j}, \\
&\quad \forall (i,j) \in \mathcal{B}, \quad \mathbf{w}^T f(\varphi[I_i]) - \mathbf{w}^T f(I_j) \geq 1 - \eta^+_{i,j}, \\
&\quad \forall (i,j) \in \mathcal{B}, \quad \mathbf{w}^T f(\varphi[I_j]) - \mathbf{w}^T f(I_i) \geq 1 - \eta^-_{i,j}
\end{aligned} \tag{9}$$

where $\eta^+_{i,j}$ and $\eta^-_{i,j}$ are errors standard in SVM formulation. Thus, we effectively resolve the ambiguity and double the number of ordering constraints from the given unordered pairs without any extra human annotation. We believe this training set enrichment scheme can be applicable to other relative problems as well as long as an enhancing filter is applicable.

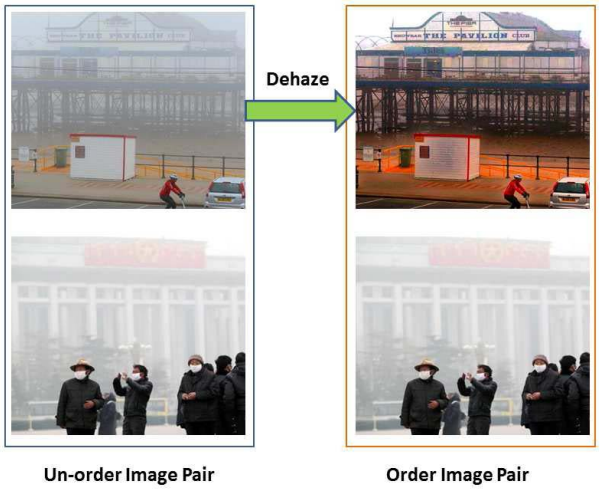

Fig. 4. New ordered pair produced from unordered pair using dehaze filter.

C. CNN-RNN Architecture

The performance of standard ranking SVM depends heavily on the design of image features (descriptors). Handcrafted features commonly used in computer vision rely on human observations and assumptions. It is however difficult to design an optimal feature to capture complex atmospheric scene visibility where a large variety of scenes has to be considered.

We propose the CNN-RNN architecture to describe the concept of scene visibility. The output is a feature vector obtained in a data-driven manner and encodes discriminative visibility information. As shown in Figure 5, our architecture consists of two deep learning modules, namely, the CNN module and the RNN model module. The CNN module describes the global view, while the RNN module simulates how humans search for the farthest discerned region. We first introduce the two modules independently, and then describe how to integrate them into a unified model by enabling information exchange among them.

1) CNN Architecture: Recent research has demonstrated the power of data-driven CNN features which consistently outperform handcrafted features thanks to the rich information inherent in the large-scale image data collection. In this paper, we adopt the CNN architecture following the design of [22] to describe global visibility of an input image (see Figure 5(a)). The CNN architecture has 7 layers. The first 5 layers are convolution layers with max-pooling with the 6^{th} and 7^{th} layers being fully connected layers. The 7^{th} layer outputs a 4096D CNN feature. Denoting θ_l as the CNN parameters of the l^{th} layer, $\Theta = \{\theta_1, \ldots, \theta_7\}$ is the set of CNN parameters.

2) RNN Model: We now model how to search for the farthest discerned region using the RNN model. As shown in Figure 5(b), starting with the whole image, the coarse-to-fine region will gradually zoom into the farthest discerned region in $6K+1$ states in a sequential manner (see Figure 2)

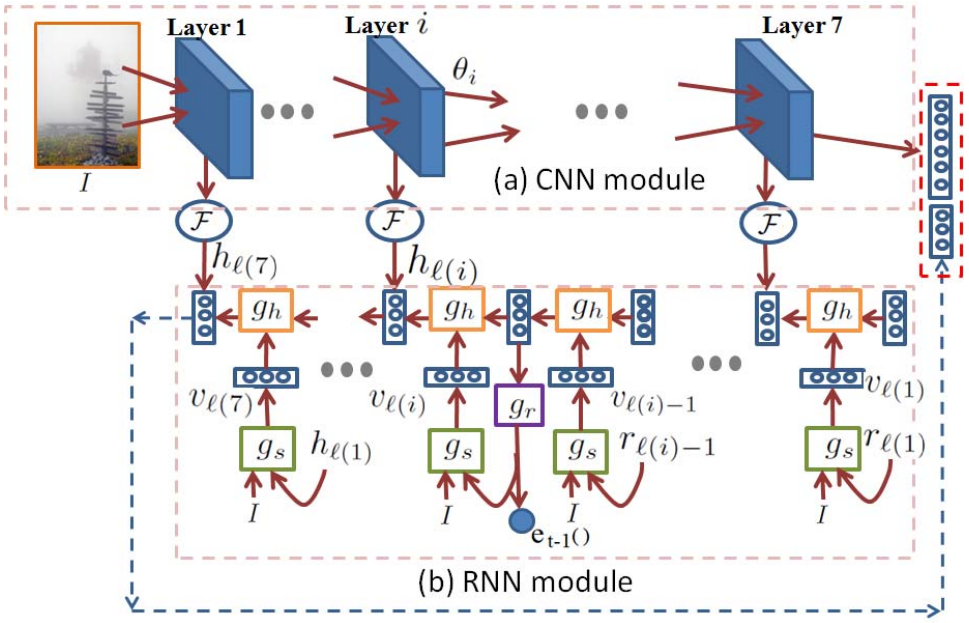

Fig. 5. CNN-RNN Architecture. (a) and (b) are respectively the CNN and RNN model. Red dashed line box is the output feature. We use $c = 8$.

where $K = 3$ is used in our paper, since we found 20 to 30 states are sufficient to capture the attention shift. The choice of $6K + 1$ is to make sure every K layers correspond to a CNN layer.

In the training phase, the ground truth coarse-to-fine attention regions are produced as follows. Coarse-to-fine regions in the first state and last state are respectively the whole image and the farthest discerned region, the latter of which was marked during the annotation step. For the in-between $6K - 2$ states, we assume that the attention coarse-to-fine regions are uniformly located between them. That is, their bounding box coordinates (top-left and bottom-right) are uniformly sampled between the whole image and the farthest discerned region. We train the RNN model on the labeled sequence. In the testing phase, we predict the regions without the labeled sequence. The RNN model has following three components:

a) Sensor network: At each step t, the system receives a predicted coarse-to-fine region r_{t-1} and input image I. We crop the coarse-to-fine region as the input of sensor network. Denote the sensor network as

$$v_t = g_s(c(I, r_{t-1}); \psi_s) \tag{10}$$

where ψ_s is network parameter. $c(I, r_{t-1})$ **is the region** r_{t-1} **in** I. The network we adopted is the AlexNet architecture, and the output of v_t is a 4096D feature vector which is the 6^{th} layer of the sensor network.

b) Internal state: Our RNN coarse-to-fine model maintains the internal states which presents information extracted from historical observations. At each step t, we have a state vector h_t (256D) encoding human knowledge on the search of the farthest discerned region from state 1 to state t. They are predicted by the previous state h_{t-1} and the current region visual descriptor v_t. This internal state is formed by the hidden units h_t of the recurrent neural network, and is updated over time by the core network $h_t = g_h(h_{t-1}, v_t, \psi_h)$ which is defined as

$$\begin{aligned} h_t &= g_h(h_{t-1}, v_t, \psi_h) \\ &= ReLU(\mathbf{W}_v v_t + \mathbf{W}_h h_{t-1} + d) \end{aligned} \tag{11}$$

where $\psi_h = \{\mathbf{W}_v, \mathbf{W}_h, d\}$ is the parameter of the internal state and $ReLU$ is the rectified linear unit (ReLU) activation function which computes $\max(x, 0)$.

c) Location network: The state h_t is used to predict the next location r_t by the network $r_t = g_r(h_t, \psi_r)$. We use a two-layered network to predict regions. The first layer is a hidden layer with a 128D parameter and the second layer outputs a 4D vector indicating the coordinates of the bounding boxes of the regions. In the training phase, we introduce a location L_2 norm loss function to measure the performance of the prediction of the region

$$E(\psi_r) = \sum_t e_t(\psi_r) \tag{12}$$

where $e_t(\psi_r) = \|g_r(h_t, \psi_r) - v_t\|_2^2$, which will be minimized in the unified objective function. We denote the parameters set of RNN model as $\Psi = \{\psi_s, \psi_h, \psi_r\}$.

3) Overall Architecture: **To gradually search the farthest discerned region, RNN learns an attention window sequence from the whole image to the farthest discerned region as shown in Figure 2. Therefore, the early states of RNN capture more global information, while later ones describe local information surrounding the farthest discerned region. For the CNN model, during the forward propagation, the neuron in the later CNN layer represents a larger region due to the convolution operation. Therefore, we can associate RNN state vector with the CNN layer that has similar representation scale to advance**

the representation. Inspired by [23], it is achieved by making use of shortcuts connection, which is a simple add operation. Specifically, we build shortcuts connection between the $(c-i)^{th}$ layer of CNN, where $c=8$, and the $[(i-1)K+1]^{th}$ state of the RNN model, where $i=1,\ldots,7$, as shown in Figure 5(a). For example, for $i=1$, both the 7^{th} CNN layer and the 1^{st} RNN state describe the whole image. Since the CNN layer and RNN state vector have different dimension, we introduce operation of $\mathcal{F}$ on $\theta_{c-(t-1)/K-1}$ with regard to attention region r_t. In detail, the operation of $\mathcal{F}(\theta_{c-(t-1)/K-1}, r_t)$ is to extract the parameters of $\theta_{c-(t-1)/K-1}$ that present the pixels in the region of r_t, then, uniformly sample 256 parameters from these parameters. Note that the sampling scheme we use is max-pooling in a non-overlapping manner. Therefore, we design our shortcuts connection as,

$$h'_t = h_t + \mathcal{F}(\theta_{c-(t-1)/K-1}, r_t) \tag{13}$$

The back-propagation is applied on h'_t, therefore, $\theta_{c-(t-1)/K-1}$ and h_t are directly optimized together.

D. Objective Function

By embedding our CNN-RNN model into the relative learning framework, our overall objective function of relative CNN-RNN model can be written as

$$\begin{aligned}\min_{\mathbf{w},\Phi} \|\mathbf{w}\|_2^2 + C[&\sum_{(i,j)\in\mathcal{O}} \xi_{i,j} + \sum_{(i,j)\in\mathcal{U}} \gamma_{i,j} + \sum_{(i,j)\in\mathcal{B}} \eta^+_{i,j} + \eta^-_{i,j}] \\ &+\kappa E(\psi_r) \qquad (14)\\ s.t.\ \forall(i,j)\in\mathcal{O},\quad &\mathbf{w}^T[f(I_i,\Phi) - f(I_j,\Phi)] \geq 1-\xi_{i,j}\\ \forall(i,j)\in\mathcal{U},\quad &|\mathbf{w}^T[f(I_i,\Phi) - f(I_j,\Phi)]| = \gamma_{i,j}\\ \forall(i,j)\in\mathcal{B},\quad &\mathbf{w}^T f(\varphi[I_i],\Phi) - \mathbf{w}^T f(I_j,\Phi) \geq 1-\eta^+_{i,j}\\ \forall(i,j)\in\mathcal{B},\quad &\mathbf{w}^T f(\varphi[I_j],\Phi) - \mathbf{w}^T f(I_i,\Phi) \geq 1-\eta^+_{i,j} \qquad (15)\end{aligned}$$

where $f(I_i,\Phi)$ is the output feature of the CNN-RNN model ((4096+256)-dimensional), $\Phi=\{\Psi,\Theta\}$ is the set of parameters, and κ is a balance weight. $E(\psi_r)$ is the coarse-to-fine attention region regression error function introduced in Eq. (12).

E. Model Learning

We adopt an iterative optimization scheme to solve the parameters. That is, we iteratively optimize $\mathbf{w}$ and Φ by fixing one and optimizing the other at each iteration. For $\mathbf{w}$ optimization, with fixed Φ, the optimization problem is a standard ranking SVM with stable solver available. We use the solver provided by [24]. Here we focus on how to optimize Φ given fixed $\mathbf{w}$.

To simplify the notation, we denote $q_{i,j}(\Phi) = f(I_i,\Phi) - f(I_j,\Phi)$ and $p_{i,j}(\Phi) = f(\varphi[I_i],\Phi) - f(I_j,\Phi)$. With fixed $\mathbf{w}$, the objective function to be minimized is

$$\begin{aligned}\min_{\Phi} &\sum_{(i,j)\in\mathcal{O}} \xi_{i,j} + \sum_{(i,j)\in\mathcal{U}} \gamma_{i,j}\\ &+ \sum_{(i,j)\in\mathcal{B}} (\eta^+_{i,j} + \eta^-_{i,j}) + \kappa E(\psi_r) \qquad (16)\\ &\forall(i,j)\in\mathcal{O},\quad \mathbf{w}^T q_{i,j}(\Phi) \geq 1-\xi_{i,j} \qquad (17)\\ &\forall(i,j)\in\mathcal{U},\quad |\mathbf{w}^T q_{i,j}(\Phi)| = \gamma_{i,j} \qquad (18)\\ &\forall(i,j)\in\mathcal{B},\quad \mathbf{w}^T p_{i,j}(\Phi) \geq 1-\eta^+_{i,j} \qquad (19)\\ &\forall(i,j)\in\mathcal{B},\quad \mathbf{w}^T p_{j,i}(\Phi) \geq 1-\eta^-_{i,j} \qquad (20)\end{aligned}$$

For constraint Eqs. (17), (19) and (20), the error $\xi_{i,j}$ and $\eta^+_{i,j}$, $\eta^-_{i,j}$ can be expressed as their respective hinge loss functions

$$J^o_{i,j}(\Phi) = \{0, 1-\mathbf{w}^T q_{i,j}(\Phi)\},\quad \forall(i,j)\in\mathcal{O} \tag{21}$$
$$J^b_{i,j}(\Phi) = \{0, 1-\mathbf{w}^T p_{i,j}(\Phi)\},\quad \forall(i,j)\in\mathcal{B} \tag{22}$$
$$J^b_{j,i}(\Phi) = \{0, 1-\mathbf{w}^T p_{j,i}(\Phi)\},\quad \forall(i,j)\in\mathcal{B} \tag{23}$$

For $\gamma_{i,j}$, according to constraint (18), the corresponding error function is

$$J^u_{i,j}(\Phi) = |\mathbf{w}^T g_{i,j}(\Phi)|,\quad \forall(i,j)\in\mathcal{U}. \tag{24}$$

Therefore, our overall loss function can be expressed as

$$\begin{aligned}J(\Phi) = &\sum_{(i,j)\in\mathcal{O}} J^o_{i,j}(\Phi) + \sum_{(i,j)\in\mathcal{U}} J^u_{i,j}(\Phi)\\ &+ \sum_{(i,j)\in\mathcal{B}} (J^b_{i,j}(\Phi) + J^b_{j,i}(\Phi)) + \kappa E(\psi_r). \qquad (25)\end{aligned}$$

We solve $J(\Phi)$ by standard gradient descent which involves iterative forward propagation and back propagation. In forward propagation, we compute the feature along the network flow. Here we discuss how to implement back-propagation in our framework. In back-propagation, we should compute the derivative of the objective function. Similar to many effective deep learning solvers, we apply their sub-gradients in the back-propagation,

$$\frac{\partial J^o_{i,j}(\Phi)}{\partial q_{i,j}} = \begin{cases}-\mathbf{w}^T & \text{if } 1 \geq \mathbf{w}^T q_{i,j}(\Phi),\\ 0 & \text{if otherwise.}\end{cases} \tag{26}$$

$$\frac{\partial J^u_{i,j}(\Phi)}{\partial q_{i,j}} = \begin{cases}\mathbf{w}^T & \text{if } \mathbf{w}^T q_{i,j}(\Phi) > 0,\\ 0 & \text{if } \mathbf{w}^T g_{i,j}(\Phi) = 0,\\ -\mathbf{w}^T & \text{if } \mathbf{w}^T q_{i,j}(\Phi) < 0.\end{cases} \tag{27}$$

The forms of $\frac{\partial J^b_{i,j}(\Phi)}{\partial p_{i,j}}$ and $\frac{\partial J^b_{j,i}(\Phi)}{\partial p_{i,j}}$ are similar to $\frac{\partial J^o_{i,j}(\Phi)}{\partial q_{i,j}}$. For $E(\psi_r)$, we have the closed form of its gradient which is used in the back-propagation.

Extensive experiments (e.g., [22]) have shown that deep learning models using sub-gradient can also work as well as those with exact gradient. In the experimental section we will demonstrate the effectiveness of our solver. The back-propagation is iterated until the error of Eq. (14) converges.

Algorithm 1 Relative CNN-RNN Model Solver

Input: Training images $I_1, \ldots, I_n$
Initialization Φ according to section IV-F.
$i = 0$
repeat
  Fix $\mathbf{w}$, solve for Φ according to [25].
  Fix Φ, solve for $\mathbf{w}$ according to section IV-E.
  $i = i + 1$
until convergence or $i > \tau$
Output: $\mathbf{w}$ and Φ.

F. Initialization

We detail here the initialization of $\Phi = \{\Psi, \Theta\}$. For the CNN part, if Θ is initialized by random variables, it will take a long time for the deep learning solver to converge. To initialize this part, we first use the page rank algorithm [25] to sort our training data. Here, the image and order preference relationship respectively correspond to the page and link for the page rank algorithm. Then, we segment the sorted training data into 20 groups. Images in the same group have similar visibility condition. In doing so, the initial network parameters can be basically discriminative toward visibility information. Now, we can use the traditional CNN to learn a classifier on the 20 groups with the pre-trained model. We pre-train the model using Imagenet 2012 dataset, since it captures many essential properties of images. Finally, the parameters are directly used as our initial parameters. For the RNN model part Ψ, we train a RNN model without connection with CNN layers.

Our solver is outlined in Algorithm 1. Similar to typical deep learning solvers, Algorithm 1 runs until convergence or the maximum number of iterations τ has reached.

G. Normalization Mapping

For visibility estimation, our SVM output scores (ranging $[-1.53, 0.31]$) indicate the relative relationship. Given an output SVM score one cannot easily judge the visibility condition of the corresponding image during the testing phase. Therefore, we use a mapping function that maps the SVM score to [0,1] according to its percentile position in the training data. For example, if the testing score is larger than 63% of training data, the visibility score is 0.63. We record this mapping operator as ρ which is simply a lookup table. So, our final visibility score for a testing image is $\rho[\mathbf{w}^T f(I_i, \Phi)]$, given the learned parameters $\mathbf{w}$ and Φ.

H. Transfer to Absolute Visibility Prediction

As aforementioned, image sets with absolute visibility meter readouts are difficult to acquire and typically comes in small scale. Implementing regression directly on small-scale dataset will inevitably prone to over-fitting. To deal with the problem, we instead propose to fine tune the above learned relative model.

Denote I_i and y_i as the i^{th} image and its absolute visibility. Similar to the definition of Eq. (1), our regression function is expressed as,

$$h(\mathbf{z}^T, \Phi; I) = \mathbf{z}^T u(I, \Phi) + b, \tag{28}$$

where $u(\Phi; I)$ has the same structure of $f(I, \Phi)$ which outputs a 4096D feature, $\mathbf{z}$ and b are respectively the weight and bias term. The following model is used to learn $\{\mathbf{z}, b\}$ and ϕ

$$\begin{aligned} \min_{\mathbf{z},\Phi} \ & \|\mathbf{z}\|_2^2 + C\sum_i(\xi_i^+ + \xi_i^-) + \kappa E(\psi_r) \\ s.t. \ & \mathbf{z}^T u(I_i, \Phi) + b - y_i \geq 1 - \xi_i^+, \\ & y_i - \mathbf{z}^T u(I_i, \Phi) - b \geq 1 - \xi_i^-, \\ & \xi_i^+, \quad \xi_i^- \geq 0 \end{aligned} \tag{29}$$

We learn $\{\mathbf{z}, b\}$ using support vector regression (SVR). For learning the network parameter Φ, we fine tune the network parameters on the learned relative model, and the back-propagation procedure is similar to the above. Then, we iteratively learn $\{\mathbf{z}, b\}$ and Φ until convergence. Thanks to the learned relative model, only a small number of iterations suffices for $\{\mathbf{z}, b\}$ and Φ to converge to a good solution.

Fig. 6. Our regressed visibility value ($\widehat{y}$) and the ground truth (y^o). The red circles indicate the landmark for visibility measurement.

V. Experiments

In this section, we first describe our experimental and evaluation settings. Then, we compare our performance with a number of baseline solutions, followed by an analysis of our solver. A user study is then presented where environmental scientists were involved. We will discuss our new relative learning metric – relative AUC. Comparative results will be benchmarked using this new metric. Finally, we describe the performance of absolute visibility of the regressed model.

A. Evaluation and Comparison

1) Evaluation Setting: In evaluating our relative visibility, the dataset is partitioned into two sets. A total of 30,000 images were used for training, while the other 7,634 images were used for testing. We partition the two

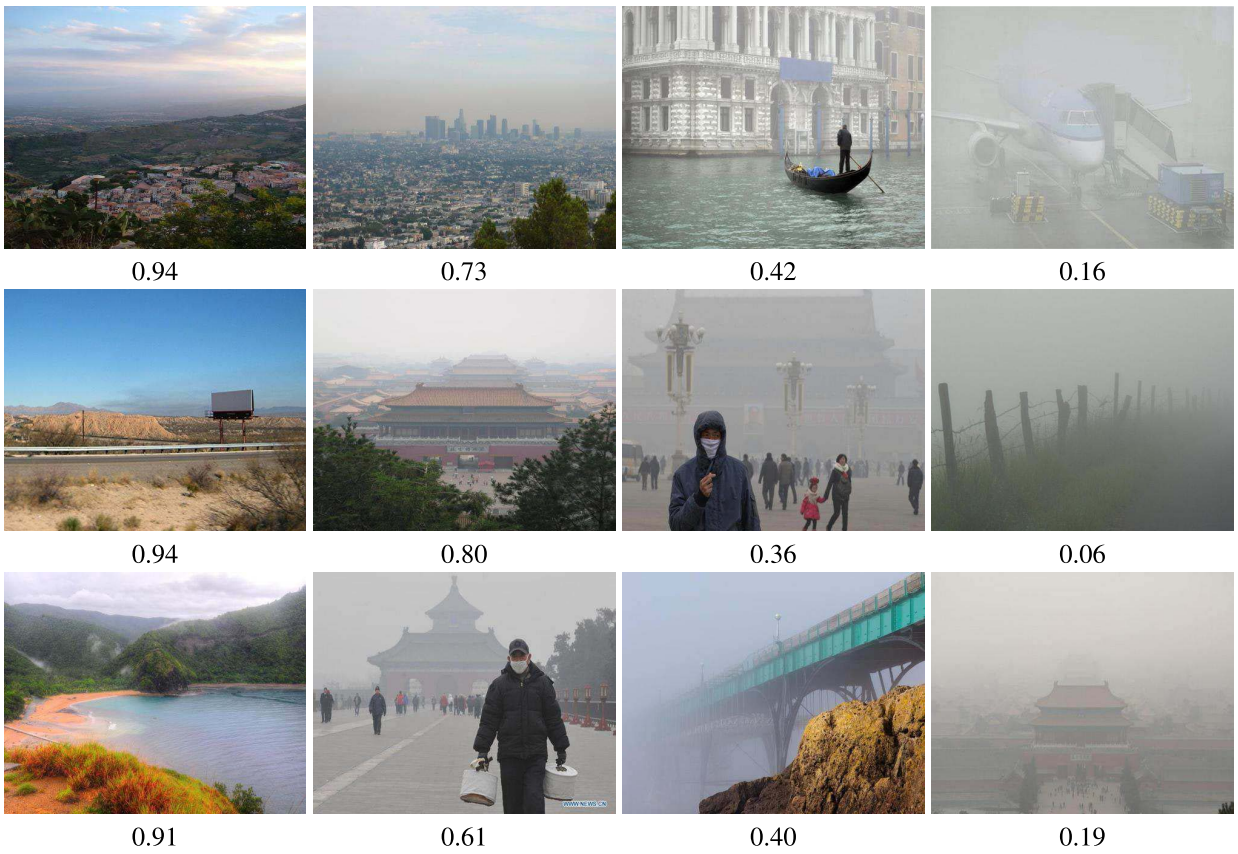

Fig. 7. Estimated relative visibility are shown for the above images which are used in our user studies.

sets by random sample selection to enable cross validation. For each image we randomly select three other images in the same image set (training or testing) to build a min-set (that is, 4 images). All pairwise visibility relationships are labeled, that is, 6 visibility relationships for each min-set.

Therefore, we have 180,000 $(30,000 \times 6)$ and 45,000 $(7,634 \times 6)$ human responses respectively for training and testing. That is, we add 53.4% training data (with ordered image pairs). The unordered images pairs constitute 25.8%

We learn a visibility model $\{\mathbf{w}, \Phi\}$ on the training data, and then apply the learned parameters on the testing data. We obtain the visibility score of each testing comparison pair.

2) Relative AUC: In [24] the evaluation of relative learning problem in computer vision was described. Applying the evaluation in our case: if $\{\rho[\mathbf{w}^T f(I_i, \Phi)] > \rho[\mathbf{w}^T f(I_j, \Phi)]\}$, we predict $i \succ j$ in visibility; else $i \prec j$. The predictions are then compared to the ground-truth relative ordering.

However, the above relative learning evaluation metric [24] have two shortcomings. First, it is in fact binary classification with 50% accuracy baseline that can be achieved simply by random guess. Non-confident predictions may by chance be counted as correct by such binary decision. Second, unordered pairs in the testing data should also contribute to the performance of output relative score, since good predictor would ensure outputting similar scores for two elements in an unordered pair.

Therefore, we propose a new *relative AUC* metric to address these two shortcomings, where AUC is the abbreviation of Area Under Curve. Let τ be a threshold or tolerance for equal scores, that is, if the score difference in a pair is smaller than τ, we label this pair as an unordered pair; otherwise, we will report it as an ordered pair with the corresponding order label. In doing so, both the ordered and unordered pairs contribute to the evaluation. Given this threshold scheme, precision-recall curves can be drawn by varying the threshold τ. The area under the curve is our relative AUC. We report the mean result of 5 rounds cross validation.

3) Comparison With Baseline Solutions: In the absence of representative systems with similar functions, we compare our method with a number of baseline solutions, including the low-level vision haze transmission estimation, relative SVM with low-level features, and the use of CNN and RNN model independently.

For low-level haze transmission estimation, we adopt [13] to estimate the transmission map. Three measurement criteria are tested: the mean, median and minimum values of the transmission map as visibility score. In particular, Tang *et al.* [20] studied different features related to haze. We compute all of the feature maps resulting in a total of 13 maps. For each feature map, we produce a 64D feature by constructing spatial pyramid with 8×8 median-pooling. Then, we produced a 13×64 dimensional feature by concatenating features from the 13 maps. We name it as the Tang feature.

For the relative SVM with low-level features, we combine the haze feature (85D) [28] for atmosphere description and GIST features (512D) [29] for scene description.

TABLE I

COMPARISON WITH VARIOUS BASELINE AND RELATIVE RANKING SOLUTIONS (*Mean* $\pm$ *Variance* IN %). "RELATIVE SVM (NONLINEAR KERNEL)" REFERS TO THE USE OF SVM WITH NONLINEAR KERNEL; WE TESTED 3 KERNELS, NAMELY, POLYNOMIAL, RADIAL BASIS AND SIGMOID AND REPORT THE BEST ONE (POLYNOMIAL KERNEL). DATA AUGMENTATION REFERS TO THE USE OF DEHAZING FILTER TO SUBSTANTIALLY INCREASE THE NUMBER OF ORDERING CONSTRAINTS. "RELATIVE CNN INITIALIZATION" MEANS EXTRACTING FEATURE ON THE INITIALIZATION CNN NETWORK

	accuracy
transmission estimation (mean)	67.1 ± 1.4
transmission estimation (median)	69.5 ± 1.2
transmission estimation (min)	56.2 ± 1.1
relative SVM + haze and scene feature	72.3 ± 1.5
relative SVM + color feature	70.4 ± 1.4
relative SVM + feature of [25]	72.2 ± 1.6
relative SVM + Tang feature	73.1 ± 1.4
relative SVM (nonlinear kernel) + haze and scene feature	71.3 ± 1.5
[2] + haze and scene feature	70.6 ± 1.4
[6] + haze and scene feature	71.7 ± 1.2
overall back-propagation solution	77.1 ± 1.3
Ours (relative CNN initialization)	80.6 ± 1.0
Ours (relative CNN)	85.8 ± 1.2
Ours (relative RNN model)	82.7 ± 1.1
Ours (without shortcuts connection)	87.1 ± 1.4
Ours (without data augmentation)	85.4 ± 1.3
Ours (CNN-RNN model)	$\mathbf{90.3 \pm 1.2}$

TABLE II

COMPARISON WITH VARIOUS BASELINES UNDER THE METRIC OF RELATIVE AUC (MEAN $\pm$ VARIANCE)

	accuracy
transmission estimation (mean)	34.7 ± 1.1
transmission estimation (median)	32.8 ± 1.0
transmission estimation (min)	24.2 ± 1.1
relative SVM + haze and scene feature	50.1 ± 1.1
relative SVM + feature of [25]	53.1 ± 1.1
relative SVM + Tang feature	54.5 ± 1.0
[2] + haze and scene feature	51.2 ± 1.0
[6] + haze and scene feature	51.1 ± 0.9
overall back-propagation solution	68.2 ± 1.0
Ours (relative CNN initialization)	69.0 ± 1.1
Ours (relative CNN)	74.1 ± 1.1
Ours (relative RNN model)	71.2 ± 1.0
Ours (without shortcuts connection)	77.2 ± 1.1
Ours (without data augmentation)	75.7 ± 1.3
Ours (CNN-RNN model)	$\mathbf{82.2 \pm 1.1}$

We concatenate the two features to produce a 596D feature. So the combined feature is supposed to capture both the haze and scene properties at the same time.

In addition, a number of relative learning solvers are also evaluated. The mainstream ranking learning solvers are summarized and implemented in [30]. The solvers [26], [27] in [30] were modified to use the 596D haze-scene feature. We also test another naive weather related features, the color histogram feature.

Finally, we also compare three baselines: (1) the use of CNN only, (2) the use of RNN model only, and (3) combining CNN and RNN feature without bridge connection.

The results are summarized in Tables I and II which show that our method is better suited to visibility estimation. The low-level transmission estimation method does not incorporate adequate scene description ingredient. In relative SVM and other ranking learning methods with combined haze and GIST feature, the disadvantage is obvious: handcrafted features are inadequate to describe atmospheric visibility for diversified scenes. The pure CNN feature benefits our relative deep learning framework, but the solution in the top layer is less optimal and weak in modeling ranking. In comparison, our framework is empowered by the data-driven CNN-RNN features derived from the rich feature hierarchies inherent in the training data, while in the top structure we metabolize a stable ranking learning solver for relative SVM. As we can see, without enriching or augmenting our training data with the automatically generated ordering constraints, the performance drops by about 5%, verifying that our training data enrichment strategy provides more useful information. We observe that using the RNN model alone is less effective. This is because it only describes historical shift. When the RNN works in tandem with the CNN network, a performance boost is observed which shows that the two modules are complementary and benefit each other. We also observe the use of shortcuts connection can significantly outperform the trivial concatenation of CNN and RNN features.

B. User Studies

We conducted our user studies which were participated by environmental scientists to verify the effectiveness of our proposed method on atmospheric visibility estimation. We invited six environmental scientists (most have a relevant PhD degree) working in reputable observatories around the world. They have extensive experience in judging visibility from photos. They did not get paid and participated in our user studies on a voluntary basis, so we gratefully acknowledge their generous help. We asked them to score our results on a 100-point scale. The scoring instructions are: full score 100 indicates that our visibility score exactly matches with their professional judgment of the atmospheric visibility condition; a score of 80 indicates our estimated visibility is highly faithful; 60 indicates that our visibility measurement can fairly reflect the atmospheric visibility; 0 indicates that we did an awful job. We randomly selected four images from our results to build a testing set. The volunteering scientists were given a total of 20 testing sets. Figure 7 shows the data and the Table III tabulates the scores. The scores from environmental scientists are all above 80, indicating that our results are highly faithful. Our method also outperforms other baseline methods.

The limited user studies demonstrate that our method is effective for visibility estimation.

C. Absolute Visibility Estimation

We evaluate the performance of our absolute visibility estimation on our small sized dataset. We partition the 3,146 images into the training and testing datasets which consist of respectively 60% and 40% of the dataset. We learned

TABLE III

SCORING RESULTS BY 6 ENVIRONMENTAL SCIENTISTS. THE NUMBER BEFORE AND AFTER $\pm$ ARE MEAN AND VARIANCE RESPECTIVELY

expert	# 1	# 2	# 3	# 4	# 5	# 6
transmission estimation (median)	64.9 ± 6.45	67.7 ± 8.05	64.6 ± 8.10	68.4 ± 7.71	66.3 ± 7.11	69.3 ± 7.47
relative SVM + haze and scene feature	73.6 ± 5.75	74.8 ± 7.56	75.7 ± 7.33	76.9 ± 6.93	75.8 ± 8.28	72.8 ± 7.15
relative SVM + Tang feature	76.3 ± 5.60	75.1 ± 8.32	76.1 ± 7.44	75.2 ± 6.39	74.9 ± 7.27	73.3 ± 8.83
overall back-propagation solution	84.5 ± 6.01	82.7 ± 7.14	85.5 ± 8.94	85.6 ± 7.35	82.5 ± 8.31	83.7 ± 7.64
Ours	$\mathbf{87.4 \pm 6.12}$	$\mathbf{84.7 \pm 8.95}$	$\mathbf{88.2 \pm 8.29}$	$\mathbf{90.1 \pm 7.30}$	$\mathbf{86.2 \pm 7.50}$	$\mathbf{87.1 \pm 8.18}$

TABLE IV

MEAN REGRESSION ERROR ON DIFFERENT METHODS. "ERROR" INDICATES MEAN REGRESSION ERROR. "SOLVING EQ. 29 WITHOUT RM" MEANS SOLVING EQ. 29 WITHOUT THE RELATIVE LEARNING MODEL. "SOLVING EQ. 29 WITH NATURAL IMAGE MODEL" MEANS SOLVING EQ. 29 BY FINE TUNING ON THE NATURAL IMAGE MODEL (LEARNED ON IMAGENET) IN THE NEURON NETWORK UPDATING STEP

	Error
SVR + haze and gist feature	0.57
Solving Eq. 30 without RM	0.62
Solving Eq. 30 with natural image model	0.45
Ours	**0.19**

an absolute visibility regressor on the training dataset and then performed testing with the testing data. We evaluate the regression error by $\frac{|\widehat{y}-y^o|}{y^o}$, where $\widehat{y}$ and y^o are respectively the regressed and ground truth visibility reading. We report the mean regression error of all testing samples. Three baseline methods are tested. First, we directly apply SVR on the haze feature + gist feature. Second, we learned Eq. (29) without fine tuning the learned relative model; rather we begin with a random initialization. Third, we solve Eq. (29) by fine tuning the neuron network on the learned natural image model in ImageNet. The results are tabulated in Table IV. Figure 6 shows sample visibility regression results. The comparison results show that our proposed solution outperforms the tested baselines. The first baseline performs quite poorly due to the fact that low-level feature fails to capture complex visibility. For the second baseline, due to the limited training data, it is difficult to fit the neuron network well with a large number of parameters. We observe that fine tuning on the natural image model is less effective since it does not encode visibility information.

VI. CONCLUSION AND FUTURE WORK

We propose the relative CNN-RNN model, where the complementary synergy of CNN and RNN modules, with the former being "local-to-global" and the latter being "global-to-local," is effectively utilized. This results in good performance in predicting relative visibility for a great variety of situations. To enrich the training data set, we automatically synthesize additional order constraints by employing a dehaze filter. Empowered by a large-scale training data repository, the CNN-RNN features are data-driven which benefit from the rich feature hierarchies inherent in the RNN-CNN architecture. This avoids the problem of handcrafted scene/haze features which fall short of adequately accommodating high variety of different outdoor scenes. Our relative model can be effectively adapted in a small data scenario where absolute visibility data are typically sparsely available. Our framework is scalable to include more data which are arguably not difficult to annotate since human and computer are better in relative judgment when it comes to visibility measurement. The visibility datasets will be released to accompany the paper in the project website. While the paper focuses on learning relative atmospheric visibility, since the input are images, we believe the ideas and techniques developed in this paper can be applied to other ranking applications that involve images. In the future we are interested in applying this new relative CNN-RNN framework in other relative attributes learning problems.

REFERENCES

[1] J. J. Koenderink, "Pictorial relief," *Philos. Trans. Roy. Soc. London A, Math., Phys. Eng. Sci.*, vol. 356, no. 1740, pp. 1071–1086, 1998.

[2] V. Mnih, N. Heess, and A. Graves, "Recurrent models of visual attention," in *Proc. NIPS*, 2014, pp. 2204–2212.

[3] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, Oct. 2016.

[4] N. Graves and S. Newsam, "Using visibility cameras to estimate atmospheric light extinction," in *Proc. WACV*, 2011, pp. 577–584.

[5] D. Bäumer, S. Versick, and B. Vogel, "Determination of the visibility using a digital panorama camera," *Atmos. Environ.*, vol. 42, no. 11, pp. 2593–2602, 2008.

[6] N. Hautiére, R. Babari, É. Dumont, R. Brémond, and N. Paparoditis, "Estimating meteorological visibility using cameras: A probabilistic model-driven approach," in *Proc. ACCV*, 2011, pp. 243–254.

[7] J. P. Oakley and B. L. Satherley, "Improving image quality in poor visibility conditions using a physical model for contrast degradation," *IEEE Trans. Image Process.*, vol. 7, no. 2, pp. 167–179, Feb. 1998.

[8] P. Carr and R. Hartley, "Improved single image dehazing using geometry," in *Proc. Digit. Image Comput., Techn. Appl. (DICTA)*, 2009, pp. 103–110.

[9] J. Kopf *et al.*, "Deep photo: Model-based photograph enhancement and viewing," *ACM Trans. Graph.*, vol. 27, no. 5, pp. 116-1–116-10, Dec. 2008.

[10] Y. Y. Schechner, S. G. Narasimhan, and S. K. Nayar, "Instant dehazing of images using polarization," in *Proc. CVPR*, 2001, p. 325.

[11] S. Shwartz, E. Namer, and Y. Y. Schechner, "Blind haze separation," in *Proc. CVPR*, 2006, pp. 1984–1991.

[12] S. G. Narasimhan and S. K. Nayar, "Vision and the atmosphere," *Int. J. Comput. Vis.*, vol. 48, no. 3, pp. 233–254, 2002.

[13] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2341–2353, Dec. 2011.

[14] Q. Yan, L. Xu, and J. Jia, "Dense scattering layer removal," in *Proc. SIGGRAPH Asia Tech. Briefs*, 2013, Art. no. 14.

[15] G. Meng, Y. Wang, J. Duan, S. Xiang, and C. Pan, "Efficient image dehazing with boundary constraint and contextual regularization," in *Proc. ICCV*, 2013, pp. 617–624.

[16] R. T. Tan, "Visibility in bad weather from a single image," in *Proc. CVPR*, 2008, pp. 1–8.

[17] R. Fattal, "Single image dehazing," *ACM Trans. Graph.*, vol. 27, no. 3, p. 72, Aug. 2008.

[18] S. G. Narasimhan and S. K. Nayar, "Shedding light on the weather," in *Proc. CVPR*, 2003, p. 665.
[19] S. Metari and F. Deschenes, "A new convolution kernel for atmospheric point spread function applied to computer vision," in *Proc. ICCV*, 2007, pp. 1–8.
[20] K. Tang, J. Yang, and J. Wang, "Investigating haze-relevant features in a learning framework for image dehazing," in *Proc. CVPR*, 2014, pp. 2995–3002.
[21] T. Joachims, "Optimizing search engines using clickthrough data," in *Proc. SIGKDD*, 2002, pp. 133–142.
[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.
[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.
[24] D. Parikh and K. Grauman, "Relative attributes," in *Proc. ICCV*, 2011, pp. 503–510.
[25] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the Web," Stanford InfoLab, Stanford, CA, USA, Tech. Rep. 1999-66, 1999.
[26] C. Burges *et al.*, "Learning to rank using gradient descent," in *Proc. ICML*, 2005, pp. 89–96.
[27] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *J. Mach. Learn. Res.*, vol. 4, pp. 933–969, Nov. 2003.
[28] C. Lu, D. Lin, J. Jia, and C.-K. Tang, "Two-class weather classification," in *Proc. CVPR*, 2014, pp. 3718–3725.
[29] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," in *Proc. IJCV*, 2001, pp. 145–175.
[30] V. Dang, "Ranklib," Univ. Massachusetts Amherst, Amherst, MA, USA, 2011. [Online]. Available: https://sourceforge.net/p/lemur/wiki/RankLib/

Yang You received the B.S. degree from Shanghai Jiao Tong University, China, in 2016 and the M.S. degree from the University of Virginia, USA, in 2017. He is currently pursuing the Ph.D. degree with the Mechanical Engineering Department, Shanghai Jiao Tong University. His research interests include computer vision, reinforcement learning, and robotics.

Cewu Lu (M'13) received the B.S. degree from the Chongqing University of Posts and Telecommunications, China, in 2006, the M.S. degree from the Graduate University of Chinese Academy of Sciences, China, in 2009, and the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Hong Kong, in 2013. He was a Research Fellow with the Hong Kong University of Science and Technology, Hong Kong. He was a Research Fellow with Stanford University, USA, from 2014 to 2015. He is currently a Research Professor with Shanghai Jiao Tong University. His research interests include activity recognition, object detection and image/video processing. He received the Best Paper Award from NPAR 2012 and served as an Associate Editor for gtCVPR journal and a reviewer for several major computer vision and graphics conferences and journals such as TPAMI and TOG.

Weiming Wang received the Ph.D. in knowledge-based engineering from the Shanghai Jiao Tong University, China. He is currently the Associate Professor with the School of Mechanical Engineering, Shanghai Jiao Tong University. His research interests include machine learning for robotics, and human–robot interaction.

Chi-Keung Tang received the M.Sc. and Ph.D. degrees in computer science from the University of Southern California, Los Angeles, USA, in 1999 and 2000, respectively. Since 2000, he has been with the Department of Computer Science, The Hong Kong University of Science and Technology, Hong Kong, where he is currently a Full Professor. He was an Adjunct Researcher with the Visual Computing Group, Microsoft Research Asia. His research interests include computer vision, computer graphics, and human–computer interaction. He was on the Editorial Board of the *International Journal of Computer Vision*. He served as an Area Chair for ICCV 2007, ICCV 2009, ICCV 2011, and ICCV 2015, and a Technical Papers Committee Member for the inaugural SIGGRAPH Asia 2008, SIGGRAPH 2011, SIGGRAPH Asia 2011, SIGGRAPH 2012, SIGGRAPH Asia 2014, and SIGGRAPH Asia 2015. He was an associate editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.