A Fast Method for Estimating Transient Scene Attributes

Ryan Baltenberger, Menghua Zhai, Connor Greenwell, Scott Workman, Nathan Jacobs
Department of Computer Science, University of Kentucky
{rbalten, ted, connor, scott, jacobs}@cs.uky.edu

Abstract

We propose the use of deep convolutional neural networks to estimate the transient attributes of a scene from a single image. Transient scene attributes describe both the objective conditions, such as the weather, time of day, and the season, and subjective properties of a scene, such as whether or not the scene seems busy. Recently, convolutional neural networks have been used to achieve state-of-the-art results for many vision problems, from object detection to scene classification, but have not previously been used for estimating transient attributes. We compare several methods for adapting an existing network architecture and present state-of-the-art results on two benchmark datasets. Our method is more accurate and significantly faster than previous methods, enabling real-world applications.

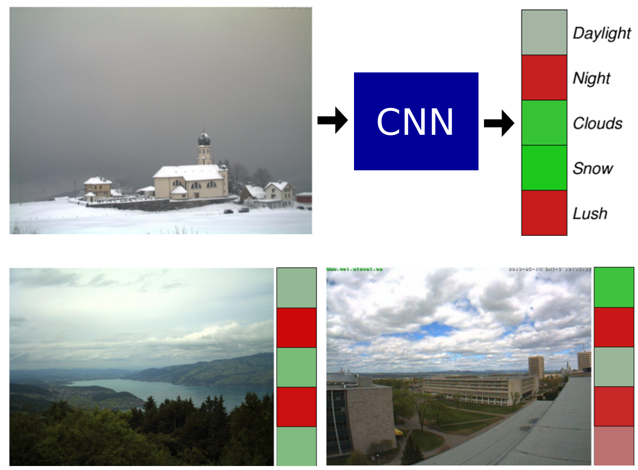


Figure 1: Our method predicts transient scene attributes from a single image using a deep convolutional neural network. For a subset of attributes, the predicted values (green=attribute present, gray=uncertain, red=attribute absent) are shown for three example images.

1. Introduction

Outdoor scenes experience a wide range of lighting and weather conditions which dramatically affect their appearance. A scene can change from rainy and brooding to sunny and pleasant in a matter of hours, even minutes. The ability to quickly understand these fleeting, or transient, attributes is a critical skill that people often take for granted. Automatically understanding such subtle conditions has many potential applications, including: improving context-dependent anomaly detection [5]; enabling attribute-oriented browsing and search of large image sets[13, 29]; estimating micro-climate conditions using outdoor webcams [9]; as a pre-processing step for higher-level algorithms for calibration [12, 31], shape estimation [4, 32], geolocalization [14, 33]; and environmental monitoring [10].

We propose a fast method for predicting transient attributes from a single image using deep convolutional neural networks (CNNs). CNNs have been used to obtain state-of-the-art results for many vision tasks, including object classification [16], object detection [8], and scene classification [36] but have not been used to estimate transient scene attributes. Our work addresses two specific problems related to estimating transient scene attributes. First, the problem of estimating whether it is sunny or cloudy [22], and second, predicting the degree to which various transient attributes are present in the scene [18]. To this end, we present two different networks and three different training initializations. Our methods achieve state-of-the-art results on two benchmark datasets and are significantly faster than previous approaches. Figure 1 shows an overview of our method.

The key contributions of this work are: 1) proposing several CNN training initializations for predicting transient attributes, 2) evaluating the proposed methods on two benchmark datasets, 3) releasing pre-trained networks for classifying transient scene attributes in a popular deep learning framework, and 4) demonstrating several applications of the networks to webcam image understanding.

1.1. Related Work

Attributes are high-level descriptions of a visual property which offer some additional semantic context for understanding an object, activity, or scene. For example, a *green* apple or a *cloudy* day. Representations based on such visual attributes have become increasingly popular in the vision community as they offer the ability to to generalize across categories. The first learning-based methods to take advantage of such high-level attributes arose for the task of object recognition [7, 20], demonstrating the power of learning by description. Many methods were quick to follow suit, with applications ranging from content-based image retrieval [28] to characterizing facial appearance [17]. Given their prowess, a significant amount of research has focused on identifying useful attributes [6] and crafting techniques to accurately detect them in images [30].

More recently, efforts have been made to adapt such attribute-based representations for outdoor scene understanding, where the appearance of a scene can change drastically over time. Patterson and Hays [25] constructed the SUN attribute dataset using crowd-sourcing techniques to identify a taxonomy of 102 scene attributes from human descriptions, designed to distinguish between scene categories. Lu et al. [22] use this dataset, along with two others, to classify images as either sunny or cloudy. Similarly, Laffont et al. [18] introduced the Transient Attributes dataset, focused instead on perceived scene properties and attributes that describe intra-scene variations. They defined 40 such attributes and presented methods for identifying the presence of those attributes as well as applications in photo organization and high-level image editing via attribute manipulation. To the best of our knowledge, we are the first to explore the application of convolutional neural networks for estimating transient scene attributes.

1.2. Background

Convolutional neural networks have been used extensively in recent years to obtain state-of-the-art results on a wide variety of computer vision problems. In this work, we focus on a particular CNN architecture, often called *AlexNet*, introduced by Alex Krizhevsky et al. [16] for single-image object classification. This network has eight layers with trainable parameters: five convolutional layers (each connected in a feed-forward manner) with pooling layers between each convolutional layer and three fully connected layers. The network parameters are selected by minimizing a softmax loss function. Essentially, the convolutional layers extract features from across the image and the fully connected layers combine these features to obtain a score for each possible class. The final classification decision is obtained by choosing the class with the highest output score.

While this network architecture was originally developed for single-image object classification, it has been shown to be adaptable to other problem domains. If the new problem involves multi-class classification, all that is needed is to modify the final fully connected layer to have the correct number of output classes. Then, the network weights can be *fine-tuned* by running iterations of stochastic gradient descent on the training data for the new problem [35]. The key is to start the optimization with random weights for the new final layer and weights from an already trained network for the other layers, for example using the weights from the original *AlexNet* [16], as an initial condition. If there is a large amount of training data available for the new domain, it is also possible to train the network from scratch by randomly initializing all weights [36]. For regression problems, the loss function is usually changed, often replacing the softmax loss with an L_2 loss.

2. Estimating Transient Attributes with CNNs

We propose the use of deep convolutional neural networks for estimating transient scene attributes. We develop networks for two single-image problems: the classification problem of estimating whether it is sunny or cloudy and a collection of regression problems for representing the degree to which a large number of transient attributes exist in the scene. For each of these problems, we use three different networks as starting conditions for optimization, resulting in a total of six networks. For both problems, we use the *AlexNet* CNN architecture, described in the previous section. The remainder of this section describes how we estimate network weights for each of these networks.

CloudyNet: For the problem of classifying whether an image is sunny or cloudy, we use the data provided by Lu et al. [22] to train our network, which we call CloudyNet. The dataset contains 10 000 images collected from the SUN Database [34], the LabelMe Database [27], and Flickr. Each image is assigned a ground-truth binary label, sunny or cloudy, by a human rater. We convert *AlexNet* into CloudyNet by modifying the network architecture; we update the final fully connected layer to have two output nodes.

TransientNet: For the more challenging problem of estimating the presence of a broad range of attributes in an image, we use the dataset introduced by Laffont et al. [18]. The dataset contains images from outdoor webcams in the Archive of Many Outdoor Scenes [13] and the Webcam Clip Art Dataset [19]. The webcams span a wide range of outdoor scenes, from urban regions to wooded, mountainous regions. Each webcam has 60–120 images captured in a wide range of conditions at different times of the day and on different days of the year. The final dataset consists of 8 571 high resolution images from 101 webcams. The authors define a set of 40 transient attributes, each of which

Table 1: Two class weather classification accuracy.

Method	Normalized Accuracy
Lu et al. [22]	53.1 ± 2.2
CloudyNet-I	85.7 ± 0.5
CloudyNet-P	86.1 ± 0.6
CloudyNet-H	**87.1** ± 0.3

Table 2: Transient attribute prediction errors.

Method	Average Error
Laffont et al. [18]	4.2%
TransientNet-I	4.05%
TransientNet-P	3.87%
TransientNet-H	**3.83%**

is assigned a value between zero and one, representing the confidence of that attribute appearing in an image. We modify the *AlexNet* network architecture by changing the final fully connected layer to have 40 output nodes, one for each transient attribute, and updating the loss function to an L_2 loss. We call the resulting network TransientNet.

Network Training: For each network architecture, we start the training procedure from three different initial conditions, resulting in six distinct sets of network weights. The first set of initial conditions were taken from a network that was was trained for object classification on 1.2 million images with 1000 object class from the ImageNet ILSVRC-2012 challenge [26]. We call the networks that result from this fine-tuning process CloudyNet-I and TransientNet-I. The second set of initial conditions were taken from a network [36] that was trained for scene classification on 2.5 million images with labels in 205 categories from the Places Database [36]. We call the resulting networks CloudyNet-P and TransientNet-P. The final set of initial conditions were taken from a network [36] that was trained for both object and scene classification. This hybrid network was trained on a combination of the Places Database [36] and images from the training data of ILSVRC-2012 challenge [26]. The full training set contained 205 scene categories from the Places Database and 978 object categories from ILSVRC-2012 containing about 3.6 million images. We call the resulting networks CloudyNet-H and TransientNet-H.

Implementation Details: Our networks are trained using the Caffe [15] deep learning framework, the CaffeNet reference network architecture (a variant of *AlexNet*), and pre-trained networks from the Caffe Model Zoo [1]. The full network optimization definition, the final network weights, and the output from our methods are available on the project webpage (http://cs.uky.edu/~rbalten/transient).

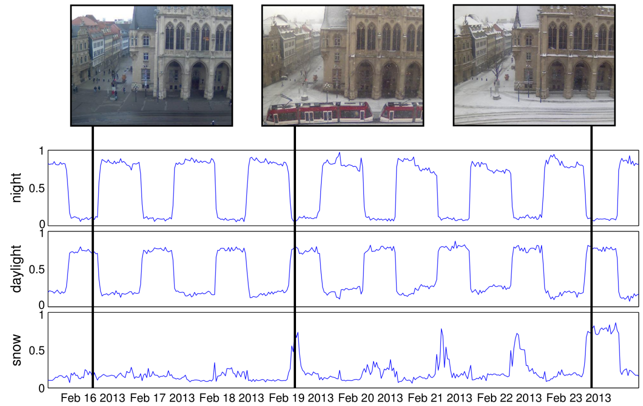


Figure 2: A snapshot of three attributes over a week of webcam data. The highlighted images show the scene at the given point in time.

3. Evaluation

We evaluated our networks on two benchmark datasets. The results show that our proposed approaches are significantly faster and more accurate than previous methods.

3.1. Two-Class Weather Classification

We evaluated our three CloudyNet variants using the dataset created by Lu et al. [22] (introduced in Section 2). We follow their protocol for generating a train/test split: we randomly shuffle sunny/cloudy images and then select 80% of each class for training and 20% for testing. This process is repeated five times resulting in five random 80/20 splits of the data. Table 1 compares the mean normalized accuracy and variance for our networks against the previous best technique. The normalized accuracy, which is the proposed evaluation metric by Lu et al., is calculated by $\max\{((a - 0.5)/0.5), 0\}$, where a is the traditionally obtained accuracy. All three of our networks outperform the state-of-the-art for two class weather classification with CloudyNet-H predicting the most accurately.

3.2. Transient Attribute Estimation

We evaluated our three TransientNet variants on the dataset created by Laffont et al. [18]. We use the same hold-out train/test split in which images from 81 webcams are used for training and images from a distinct set of 20 other webcams are used for testing. TransientNet-H has the lowest overall average error as shown in Table 2. TransientNet-P and TransientNet-H have similar performance, mostly due to them being pre-trained on similar sets of data.

In addition to having higher accuracy, our method is significantly faster. For a single image, we found Laffont et al.'s method takes an average of 3.486 seconds, but our method only requires 0.192 seconds, an 18x speed up.

3.3. Example Results

As qualitative evaluation, Figure 2 shows the time series of the predicted value (using TransientNet-H) for three attributes (*night, daylight, and snow*) from an AMOS [13] webcam over the period February 16th, 2013 to February 23rd, 2013. Note that no temporal smoothing was performed, these are raw per-image estimates. The inverse relationship between the *daylight* and *night* time series can be clearly seen. Figure 2 also shows images of the scene captured at different times, highlighting snowy and non-snowy periods.

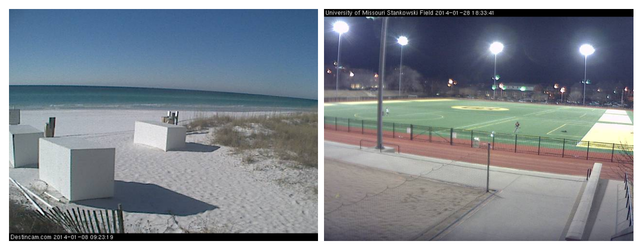
Figure 3 shows semantic average images for a single scene. Each image is the average of the 100 images with the highest score for a particular attribute. The subset of attributes shown in Figure 3 represent a wide variety of conditions of the scene. The seasonal attributes (*autumn, summer, winter*) show how the scene changes throughout the year and lighting attributes (*sunrise/sunset, daylight, night*) show the scene in various lighting conditions. Such images are easy to create and highlight the ability of our proposed technique to work across a broad range of conditions and scene types.

Figure 4 shows examples of images with an attribute that TransientNet-H mislabeled: a white-sand beach that was labeled as being a snowy image and a lit sports arena at night that was labeled as being a daylight image. Figure 5 shows examples of misclassified images using CloudyNet-H: an overcast scene of a mansion that was classified as sunny and a clear scene of a country home that was classified as cloudy.

3.4. Rapidly Labeling Sub-Images

We convert the final, fully connected layers of TransientNet to be convolutional [21] with the same number of outputs. The output from this new, "fully convolutional" network allows us to create images showing an attribute's value across an input image, as shown in Figure 6. The values for each attribute can be visualized in a single channel image. Combining three of these images results in the composite images. Figure 6a shows a composite image using the *sunny*, *lush*, and *snow* attributes as the color channels. There are no snowy areas in the input image, shown in the blue channel, and the bottom of the image contains high values for the lush attribute, shown in the green channel. The sunny attribute is higher towards the horizon and middle of the sky, shown in the red channel, possibly due to the sky being brighter in these regions. Figure 6b shows a composite image using the *sunny*, *storm*, and *snow* attributes as the color channels. The image has low values for the sunny attribute, shown in the red channel, and high values for the storm attribute, show in the green channel. The storm attribute is higher in the overcast sky towards the top of the composite image. The snow covered ground appears in the blue channel with high values for the snow attribute around the middle of the image and a dark spot corresponding to the waterway in the scene.

Figure 3: The average of the 100 most confident images for a subset of transient attributes from a given webcam.

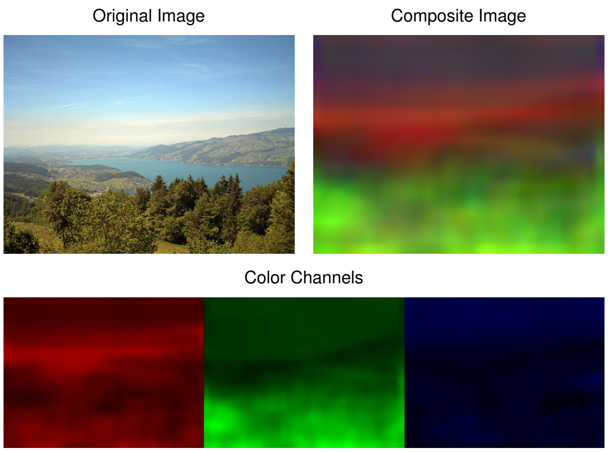


(a) Mislabeled *snow* (b) Mislabeled *daylight*

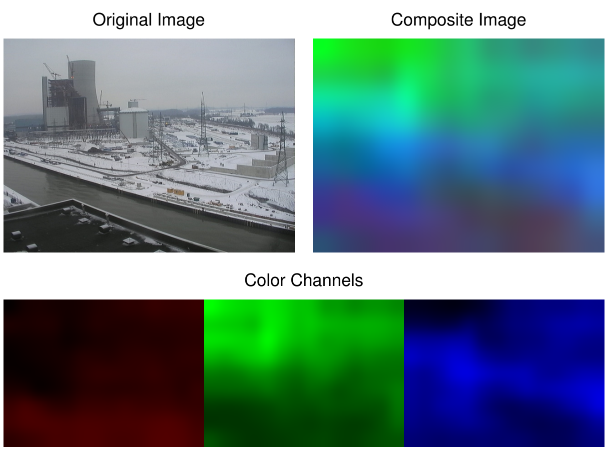
Figure 4: Two failure cases using TransientNet and their mislabeled attribute.

(a) Misclassified as *sunny* (b) Misclassified as *cloudy*

Figure 5: Two failure cases using CloudyNet and their misclassified class.

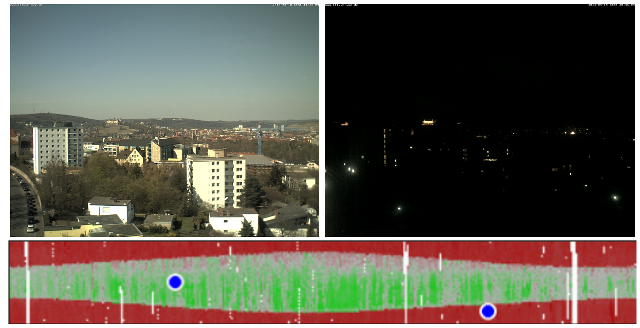


(a) RGB = [*sunny, lush, snow*]

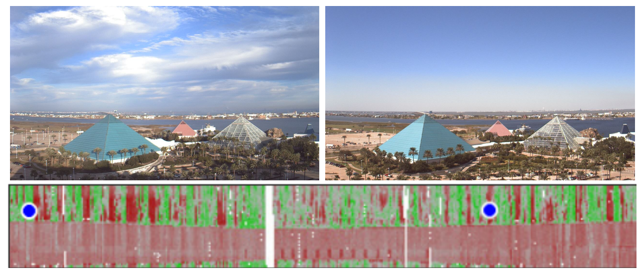


(b) RGB = [*sunny, storm, snow*]

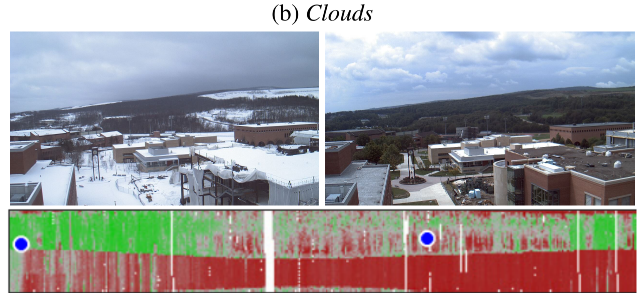
Figure 6: Composite images generated using the fully convolutional TransientNet-H. Brighter areas in each of the color channels indicate a higher attribute value.



(a) *Daylight*



(b) *Clouds*



(c) *Snow*

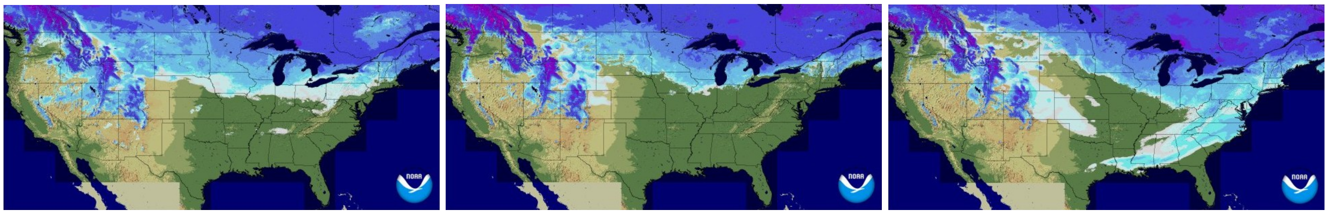
Figure 7: Example attribute summaries over a year of webcam data (green=attribute present, gray=uncertain, red=attribute absent, white=no data). The highlighted images are denoted by the blue dots within each attribute summary.

4. Applications

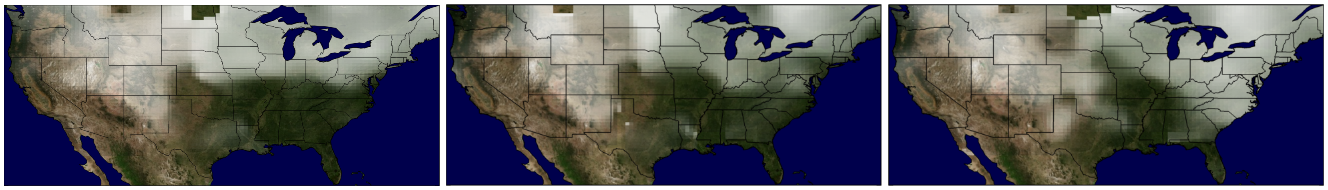
Our proposed method is both faster and more accurate than previous methods, and has potential application to many real-world problems. Here we explore applications to webcam imagery, including: 1) supporting automatic browsing and querying of large archives of webcam images, 2) constructing maps of transient attributes from webcam imagery, and 3) geolocalizing webcams.

4.1. Browsing and Querying Webcam Archives

Webcam collections such as AMOS [13] contain thousands of geolocated webcams with years of archived data. Searching for scenes, and images, with a set of desired attributes is currently a time-consuming manual process. For example, when working on outdoor photometric stereo [4], it is common to manually filter out all cloudy images. We simplify this process by using TransientNet to tag images and webcams with certain attributes. If an attribute is above a threshold (e.g., $t_h = 0.75$), the image is labeled with that attribute. The opposite is true as well. If an attribute is below a threshold (e.g., $t_l = 0.25$), the attribute is added to a list of attributes the image does not have. This enables users to find, for example, images that are both snowy and sunny using queries such as "sunny" or "not winter". Labeling is done on the image level as well as the webcam level. Attributes that are uniquely high for a webcam (i.e., $P(label|camera) \gg P(label|all\ cameras)$) are used to tag the webcam. A labeling scheme like this one allows a user to, for example, search for the snowy images from a *mysterious* webcam. This allows for easier searching of large collections of webcams.



(a) January 1st, 2014 (b) January 15th, 2014 (c) January 29th, 2014



(d) January 1st, 2014 (e) January 15th, 2014 (f) January 29th, 2014

Figure 8: Maps of the *snow* attribute from webcam data (bottom) across the continental United States in January 2014 and the corresponding map of snow depth created using remote sensing data (top) [2].

Figure 9: Map of the *snow* attribute from Figure 8f with three highlighted snowy images.

To support rapid browsing of a large webcam image collection, we create summaries of the transient attributes estimated by TransientNet. Figure 7 summarizes a year of images from AMOS webcams. Figure 7a shows one year of the *daylight* attribute, Figure 7b shows one year of the *clouds* attribute, and Figure 7c shows one year of the *snow* attribute. Each column in the summary is a single day and each row a different time of the day (in 30 minute intervals). Each pixel is colored based on the attribute value for the corresponding webcam image. Attributes such as *snow*, *cold*, and *winter* have higher values in the winter months and lower values during the summer months. The *night* and *daylight* attributes clearly show the day/night cycle for the location of the image. Properties about the scene can be inferred from these summaries. Consistently high values for the *glowing* attribute at night indicate the presence of streetlights and/or other man made light sources in the scene. Such visualizations are more robust to camera motion and more semantically meaningful than those based on PCA [11].

4.2. Mapping Weather Using Webcams

We show how to use webcams with known locations to capture the geospatial distribution of transient attributes. We downloaded data for January 2014 from 3 500 AMOS webcams across the United States. The images were labeled using TransientNet-H to create a sparse distribution of points. We then used locally weighted averaging to estimate the attribute map. This differs from the technique proposed by Murdock et al. [23, 24] in that our method uses a single model to make predictions for all cameras, while Murdock et al. create camera-specific models.

Figure 8 shows three maps for the snow attribute across the continental United States. Data from January 2014 for AMOS webcams within the continental United States and the southern edge of Canada was downloaded and labeled using TransientNet-H. These maps show predicted snow coverage using only the *snow* attribute. Variation between the three maps shows snow accumulating and melting throughout the month. Anomalous regions of high snow values, such as those along the California coast, come from false positive labels. One such region comes from a camera facing a white-sand beach, which appears visually similar to a snowy scene. Several cameras of this nature were manually pruned from the dataset. Figure 9 shows example images from selected webcams on January 29th, 2014. The first two example images show heavy snow cover in northern areas and the third example image shows the light snow cover in the south-eastern region of the United States. Maps for other attributes show expected natural phenomena (the

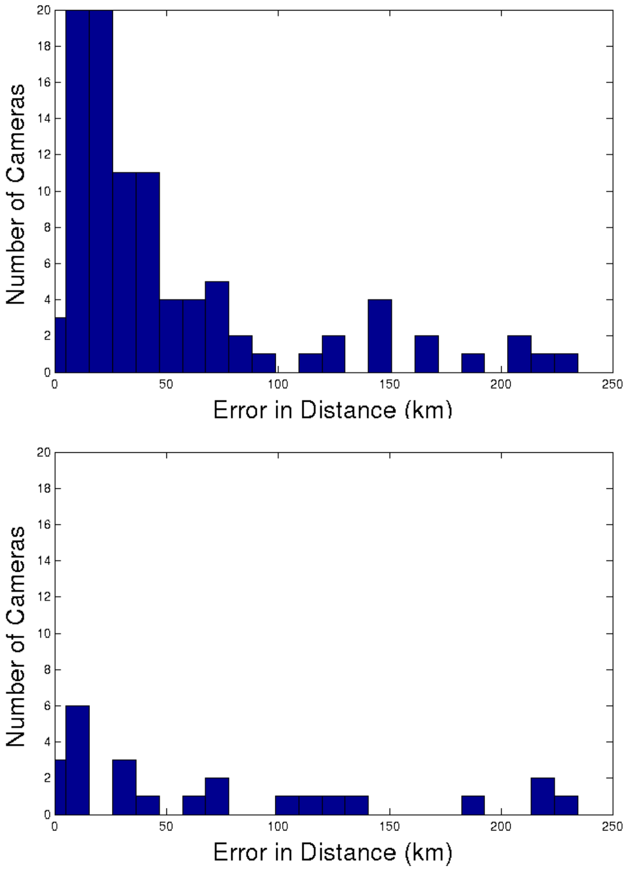


Figure 10: Webcam geolocalization errors for two methods. (top) Using the *sunny* attribute. (bottom) Using the first PCA coefficient.

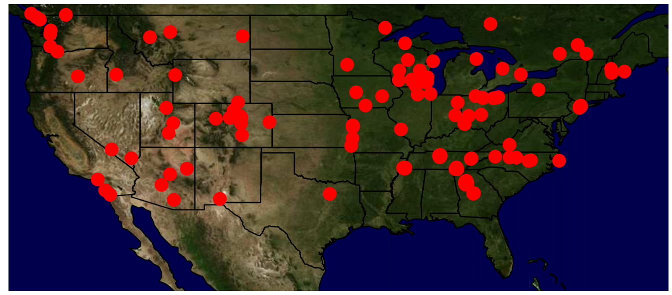


Figure 11: Distribution of webcams used in our geolocalization experiments.

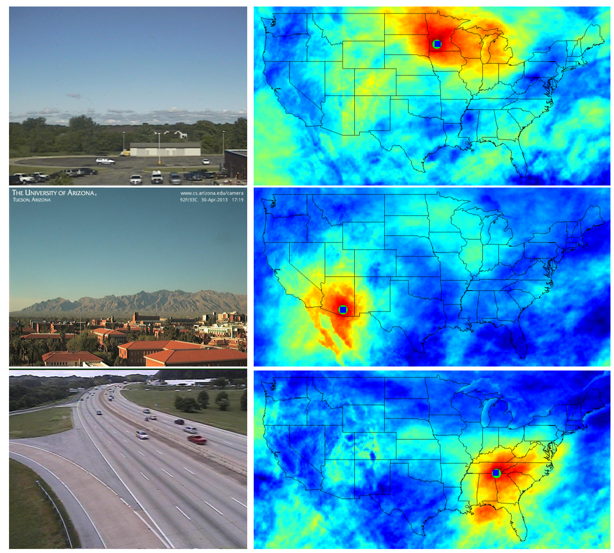


Figure 12: Webcam geolocalization results using the *sunny* attribute. (left) Webcam images and (right) estimated correlation maps, where orange means more likely. Ground-truth locations are marked by green dots, predictions by blue squares.

daylight attribute increasing/decreasing east to west as the sun rises/sets) and cues about the natural world (the *rugged* attribute higher in the mountainous west and low in the central plains).

4.3. Transient Semantics for Geolocalization

Given a sufficient amount of time, the temporal pattern of the transient attributes is a unique fingerprint of a location. Based on this observation, we propose a robust method for geolocalizing outdoor webcams. We adopt the framework of [14], in which the webcam location is found by relating temporal variations of georegistered satellite imagery and the time series of features extracted from webcam images. The estimated camera location is the center of the satellite pixel for which the intensity is most correlated with the webcam time series. The only change from the original work is replacing the PCA coefficients (which are unsupervised, but camera specific) with the transient attributes (which are supervised, but not camera specific).

For evaluation, we downloaded a year (2013) of images from 180 randomly selected webcams (Figure 11) from the AMOS dataset [13] and corresponding satellite images [3]. We found that the *sunny* attribute provided the most accurate results and use it for all further figures. Figure 12 visualizes geolocalization results for several webcams and Figure 10 shows quantitative results. Our method localizes 58% of webcams within 250km from the ground truth. As a baseline method, we repeated the experiment with the top 5 PCA coefficients. The best coefficient (the first) only locates 14% of webcams within 250km. We think the main advantage of using the transient attribute for this task is that it is less sensitive to camera jitter, a significant problem when applying PCA to outdoor webcam data. When the camera jitters it is likely that the PCA coefficients encode for motion, not changes visible in satellite imagery.

5. Conclusions

We introduced a fast method for predicting transient scene attributes in a single image. Our method achieves state-of-the-art performance on two benchmark datasets, requires no hand-engineered features, is simple to train, and is very fast at test time. In addition, it can be quickly extended to label additional attributes or adapted to new datasets with a small amount of retraining. Together, these properties make it particularly well suited to real-world applications, of which we demonstrated several.

References

[1] http://caffe.berkeleyvision.org/model_zoo.html.

[2] http://www.nohrsc.noaa.gov/nsa/index.html.

[3] http://mesonet.agron.iastate.edu.

[4] A. Abrams, C. Hawley, and R. Pless. Heliometric stereo: Shape from sun position. In *ECCV*, 2012.

[5] A. Abrams, J. Tucek, N. Jacobs, and R. Pless. Lost: Longterm observation of scenes (with tracks). In *WACV*, 2012.

[6] T. L. Berg, A. C. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 2010.

[7] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[9] M. T. Islam, N. Jacobs, H. Wu, and R. Souvenir. Images+weather: Collection, validation, and refinement. In *IEEE CVPR Workshop on Ground Truth*, 2013.

[10] N. Jacobs, W. Burgin, N. Fridrich, A. Abrams, K. Miskell, B. H. Braswell, A. D. Richardson, and R. Pless. The global network of outdoor webcams: Properties and applications. In *ACM SIGSPATIAL*, 2009.

[11] N. Jacobs, W. Burgin, R. Speyer, D. Ross, and R. Pless. Adventures in archiving and using three years of webcam images. In *IEEE CVPR Workshop on Internet Vision*, 2009.

[12] N. Jacobs, M. T. Islam, and S. Workman. Cloud motion as a calibration cue. In *CVPR*, 2013.

[13] N. Jacobs, N. Roman, and R. Pless. Consistent temporal variations in many outdoor scenes. In *CVPR*, 2007.

[14] N. Jacobs, S. Satkin, N. Roman, R. Speyer, and R. Pless. Geolocating static cameras. In *ICCV*, 2007.

[15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[17] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Describable visual attributes for face verification and image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):1962–1977, 2011.

[18] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics (SIGGRAPH)*, 33(4), 2014.

[19] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Webcam clip art: Appearance and illuminant transfer from time-lapse sequences. *ACM Transactions on Graphics (SIGGRAPH)*, 28(5), 2009.

[20] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[22] C. Lu, D. Lin, J. Jia, and C.-K. Tang. Two-class weather classification. In *CVPR*, 2014.

[23] C. Murdock, N. Jacobs, and R. Pless. Webcam2satellite: Estimating cloud maps from webcam imagery. In *WACV*, 2013.

[24] C. Murdock, N. Jacobs, and R. Pless. Building dynamic cloud maps from the ground up. In *ICCV*, pages 1–9, 2015.

[25] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge, 2014.

[27] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008.

[28] B. Siddiquie, R. S. Feris, and L. S. Davis. Image ranking and retrieval based on multi-attribute queries. In *CVPR*, 2011.

[29] L. Tao, L. Yuan, and J. Sun. Skyfinder: attribute-based sky image search. *ACM Transactions on Graphics (SIGGRAPH)*, 28(3):68, 2009.

[30] A. Vedaldi, S. Mahendran, S. Tsogkas, S. Maji, R. B. Girshick, J. Kannala, E. Rahtu, I. Kokkinos, M. B. Blaschko, D. Weiss, et al. Understanding objects in detail with fine-grained attributes. In *CVPR*, 2014.

[31] S. Workman, R. P. Mihail, and N. Jacobs. A pot of gold: Rainbows as a calibration cue. In *ECCV*, 2014.

[32] S. Workman, R. Souvenir, and N. Jacobs. Scene shape estimation from multiple partly cloud days. *Computer Vision and Image Understanding*, 134:116–129, 2015.

[33] S. Workman, R. Souvenir, and N. Jacobs. Wide-area image geolocalization with aerial reference imagery. In *ICCV*, pages 1–9, 2015.

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

[35] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014.

[36] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014.