

# Learning Geo-Temporal Image Features

Menghua Zhai<sup>1</sup>

ted@cs.uky.edu

Tawfiq Salem<sup>1</sup>

salem@cs.uky.edu

Connor Greenwell<sup>1</sup>

connor@cs.uky.edu

Scott Workman<sup>1</sup>

scott@cs.uky.edu

Nathan Jacobs<sup>1</sup>

jacobs@cs.uky.edu

Robert Pless<sup>2</sup>

pless@cse.wustl.edu

<sup>1</sup> Department of Computer Science

University of Kentucky

Lexington KY USA

<sup>2</sup> Department of Computer Science and

Engineering

Washington University

St. Louis MO USA

---

## Abstract

We propose to implicitly learn to extract geo-temporal image features, which are mid-level features related to when and where an image was captured, by explicitly optimizing for a set of location and time estimation tasks. To train our method, we take advantage of a large image dataset, captured by outdoor webcams and cell phones. The only form of supervision we provide are the known capture time and location of each image. We find that our approach learns features that are related to natural appearance changes in outdoor scenes. Additionally, we demonstrate the application of these geo-temporal features to time and location estimation.

## 1 Introduction

Outdoor images often contain sufficient visual information to understand geographic information about the scene, such as where the image was captured. Developing effective algorithms for this task has received significant attention for many years [6, 27]. The appearance of an outdoor scene can also change rapidly. These changes are often due to fleeting, or transient, attributes such as lighting and weather conditions, that dramatically affect the visual perception of an environment. For instance, consider a scene that changes from sunny and pleasant to rainy and brooding in mere minutes. Several methods have been proposed for automatically understanding and extracting these subtle characteristics from imagery [10, 13, 16, 20]. Estimating these types of transient attributes has importance in a number of applications, including: environmental monitoring [9, 26], as a pre-processing step for calibration [11, 29], and enabling semantic browsing of large photo collections [9, 13]. Our work fuses these two research areas by learning to estimate geo-temporal image features, which are related to when and where an image was captured.

Recently, a significant amount of work has explored how sources of supervision beyond manual annotation can be used to learn useful representations of images. In general, collecting manual annotations for millions, or perhaps billions, of images is prohibitively expensive. As Doersch summarizes [10], “The idea is that, given the right task, the computer can learn on its own to represent useful semantic properties of the visual world.” Such learning tasks are often referred to as *pretext tasks*; they serve as an intermediary target for learning the intended representation. For example, Doersch et al. [9] show how spatial context can be used as a supervisory signal in order to learn a visual representation for object discovery. Similarly, Pathak et al. [18] use context-based pixel prediction for pre-training a representation for classification, detection, and segmentation tasks. We extend this line of work by using time and location context to learn useful features from a large corpus of imagery.

Our work makes the following visual assumptions about the world. First, that photographs provide a direct source of context regarding the conditions under which they were captured. For example, the time of day that an image is captured is directly related to the brightness of the image (i.e., light to dark), season can indicate the expected weather conditions or how people are dressed, and location can provide evidence about anticipated styles, such as architecture. Second, these context signals are hard to extract from an image, are potentially noisy (e.g., snow in early Summer), and can be indicated by multiple sources (e.g., snow on the ground, people wearing heavy coats). These assumptions motivate our method which integrates image appearance, time, and location, the latter of which are typically recorded automatically by the imaging device.

In our approach, we explicitly model the relationship between the image, its geographic location, and the time of capture. We propose a novel convolutional neural network architecture that implicitly learns how to extract geo-temporal features from the imagery by optimizing for a set of location and time estimation tasks. Specifically, we structure our network to jointly learn feature representations for three related spaces: images, time, and location. To accomplish this, each representation, or combination of representations, is used to predict held out information. For example, the image representation and location representation (or the combination of both) are used to learn to predict when an image was captured. In total, three representations are learned using four classification tasks. We optimize all representations and tasks simultaneously, in an end-to-end fashion.

The main contributions of this work are: 1) a novel approach for learning geo-temporal image features from a large corpus of imagery without requiring image-level manual annotations; 2) an evaluation of the learned features on the task of transient attribute estimation, where our features outperform those from a network pre-trained using the strongly supervised ImageNet dataset [21]; 3) an evaluation of the accuracy of our learned estimators, highlighting the value of additional context; and 4) a novel location estimation method that uses the task of time estimation to localize a static webcam.

## 2 Related Work

Image localization, or estimating where an image was captured, is an important problem in the vision community. Typically, the problem is formulated as image retrieval using a reference database of ground-level images [6] or overhead images [15, 28, 30] with known location. Other methods have been proposed which take advantage of photometric and geometric properties such as sun position [14, 29], and many other cues. More recently, Weyand et al. [27] proposed to directly predict the geographic location of a single image using a deep

convolutional neural network by classifying the query image into a set of spatial bins. For our localization task, we adopt this classification approach and extend it to include temporal context.

Other work has explored how to estimate the time that an image was captured. Salem et al. [22] demonstrate that human appearance, including clothing and hairstyle, is a useful cue for dating images. Matzen and Snavely [17] predict timestamps for photos by matching against a time-varying reconstruction of a scene. Volokitin et al. [25] use representations extracted from CNNs to estimate ambient temperature and time of year for outdoor images. As with localization, we adopt a classification approach to estimating when an image was captured and show how these estimates improve when the image location is known.

Attribute-based representations have become popular in outdoor scene understanding to help describe how the appearance of a scene changes over time. Laffont et al. [13] introduced a taxonomy of 40 transient attributes that describe intra-scene variations along with methods for identifying the presence of such attributes in an image. Using this dataset, Baltenberger et al. [1] introduce methods for estimating the presence of transient attributes using convolutional neural networks. Jacobs et al. [9] demonstrate that principal component analysis, when applied to webcam imagery, results in a decomposition that is closely related to natural changes in the scene, including the time of day, local weather conditions, and human activity. Similarly, a body of work has sought understand local weather conditions [8, 16]. Many studies have shown that these types of transient attributes can be useful for image and camera localization tasks [11, 10].

Recent work has explored the use of self-supervision, which are sometimes referred to as pretext tasks, for training deep neural networks to capture useful visual representations [3, 18]. For example, Zhang et al. [20] show how image colorization (synthesizing colors for a grayscale image) is a powerful pretext task for learning visual representations. Pathak et al. [19] exploit low-level motion-based grouping cues for unsupervised feature learning. These methods typically exploit some known quantity of the data (e.g., pixel color values) to avoid expensive manual annotation. As a byproduct, a useful visual representation is learned. In our work, we consider two novel pretext tasks, time and location estimation.

## 3 Estimating Geo-Temporal Image Features

We propose a neural network architecture for learning geo-temporal features from images by optimizing for a set of location and time estimation tasks. An overview of the proposed architecture is shown in Figure 1. Our network takes three inputs: an image,  $I$ , the time the image was captured,  $t$ , and the location of capture,  $l$ . Each input is independently processed by a *context network* to extract mid-level features. Then, pairs of these features are used by *estimator networks* to predict distributions over time or location.

### 3.1 Context Networks

We use three *context networks*: a temporal context network,  $C_T(t)$ ; a location context network,  $C_L(l)$ ; and an image context network,  $C_I(I)$ . The output of each context network is a 128-dimensional feature with a sigmoid activation function. For the temporal context network, we parameterize the input timestamp using a one-hot encoding of month and hour of day, for a total of  $12 \times 24$  dimensions. This encoding is flattened and passed to  $C_T(t)$ , which consists of three fully-connected layers (with 256, 512, and 128 channels respectively), the

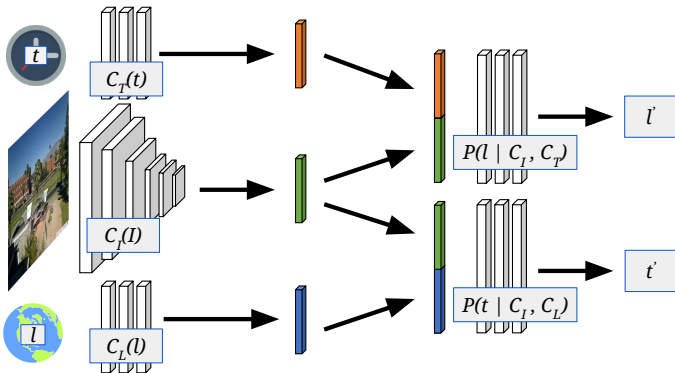


Figure 1: An overview of the proposed network architecture. Our approach learns mid-level feature representations for time (orange), location (blue), and image appearance (green) by optimizing for a set of conditional time and location estimation tasks.

first two with ReLU activations. For the location context network, we parameterize the geographic location,  $l$ , using standard 3D ECEF coordinates, which we normalize by the Earth’s radius. Other than a different input and independent network weights, the location context network is identical to the temporal context network. For the image context network, we use the *InceptionV2* architecture [23], up to the global pooling layer, to extract features. We flatten the output feature map and append the same structure as the other context networks.

### 3.2 Estimator Networks

The output of the context networks are used as input to four different estimator networks:

- *Location Estimator*,  $P(l | C_I(I))$ , which predicts location using only image features;
- *Time Estimator*,  $P(t | C_I(I))$ , which predicts capture time using only image features;
- *Time-conditioned Location Estimator*,  $P(l | C_I(I), C_T(t))$ , which predicts location using features from the image and the known capture time;
- *Location-conditioned Time Estimator*,  $P(t | C_I(I), C_L(l))$ , which predicts capture time using features from the image and the known geographic location.

Aside from different output sizes, the estimator networks have the same structure as the context networks. We discretize the output space for location and time and represent the probability as a categorical distribution (i.e., using a *softmax* activation for each estimator). For location, we use  $37 \times 72$  equal-angle “latitude  $\times$  longitude” bins. For time, we use  $12 \times 24$  “month  $\times$  hour” bins.

### 3.3 Implementation Details

We randomly initialize the *InceptionV2* network using the standard strategy [23]. We initialize all other network weights randomly using Xavier initializer [5] and simultaneously optimize them during training. For each estimator network, we have a cross entropy loss. We minimize the sum of these using the *Adam* optimizer [14] ( $\beta_1 = 0.9$  and  $\beta_2 = 0.999$ ). We use a learning rate policy that starts from 0.001 and decreases by half every 50k iterations. For regularization, we apply weight decay with rate of 0.0001. We train the proposed



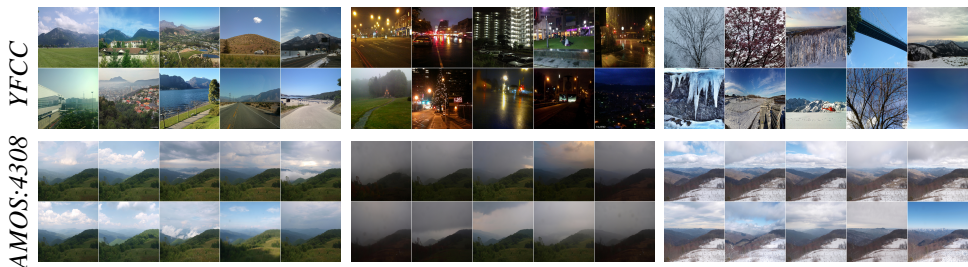


Figure 2: Relating image appearance to the image context representation by visualizing the images that yield the highest activation at three different neurons.

network for 2.5M iterations with batch size 32. We apply batch normalization [14] on every layer except the last (for both context and estimator networks). The input images are scaled to  $[-1, 1]$  and augmented by a random crop to the size of  $224 \times 224$ . We use Greenwich Mean Time (GMT) for all timestamps.

## 4 Experiments

We evaluate the context networks and estimator networks on various datasets, visualize specific features in the image context networks, and show that the image context features have strong correlations with transient image attributes.

### 4.1 Training and Evaluation Datasets

We use four main datasets to evaluate our approach. The *AMOS* dataset refers to a subset of the AMOS database [9], which is a collection of over a billion images captured from public outdoor webcams around the world. For our experiments, we use a subset of images: only from webcams with high-accuracy geolocation and images captured between 2002 to 2017. This resulted in images from 12,193 webcams from which we held-out 231 for testing. Each image has a timestamp recorded by the image collection process. The *YFCC* dataset refers to a subset of the Yahoo 100 million dataset [24], only including geotagged images from smart phones. We restricted the dataset to smart phone images since we found that non-phone images often had inaccurate timestamps. We filter out indoor images using the *Places* network [52]. This results in a training set of 892,662 images and a test set of 170,994 images. The *Hybrid* dataset refers to a combination of the *AMOS* and *YFCC* training sets (sampling equally for each mini-batch). The *TA* dataset refers to the Transient Attributes Dataset [13], which contains 8,571 images, each manually annotated with 40 transient attributes, such as sunny and cloudy.

### 4.2 Understanding the Image Context Representation

We conducted several experiments to relate image appearance to the representation learned by the image context network. To begin, we examined images that correspond with extremal activations. For this experiment, we used 10,000 images randomly sampled from the *YFCC* dataset and 7,732 images covering the year of 2015 from one webcam (ID: 4308) in the

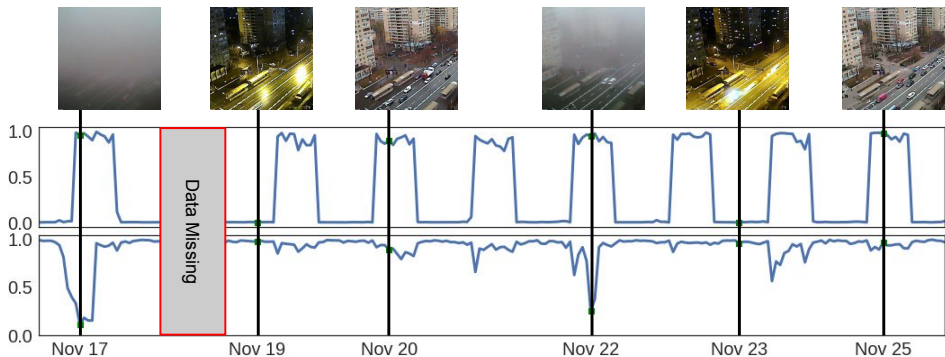


Figure 3: The time series of two neurons for a week of webcam imagery, with images showing the scene at various points. It appears that the top neuron is related to the diurnal cycle and the bottom is related to fogginess.

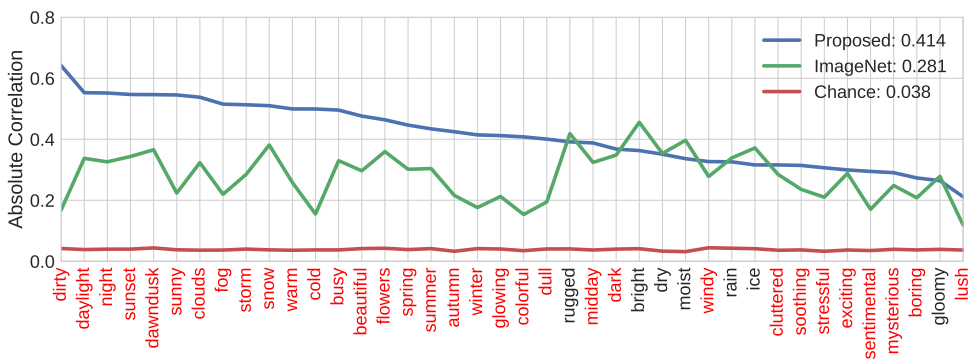


Figure 4: Maximum absolute cross-correlation scores between transient attributes and three image feature representations. For a majority of attributes our proposed representation has at least one feature that is more highly correlated than any in both baseline representations.

*AMOS* dataset. For each neuron of the image context representation, we selected the 10 images that result in the highest activation from the two different sets of images. Figure 2 shows a montage of images for three neurons. The neurons appear to capture semantically meaningful attributes, such as daylight, rainy, and winter. Similarly, we selected two neurons and visualized their signal over time for images from the webcam. Figure 3 shows how scene appearance changes are related to the image context features. For the example shown, it appears that these neurons are related to daylight and fogginess. These experiments provide evidence that the mid-level representation captured by the image context network are related to static and transient scene attributes.

### 4.3 Analyzing Feature Correlation with Transient Attributes

To analyze quantitatively how much our model learns about transient attributes, we compute the cross correlations between a mid-level representation of the image context network and the corresponding transient attribute labels of all test images in the *TA* dataset. As a baseline,

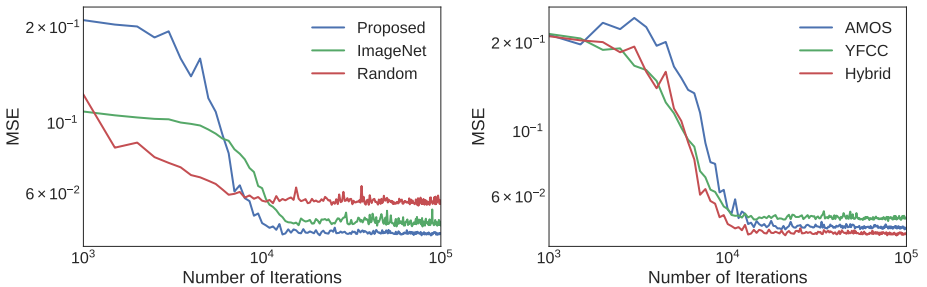


Figure 5: Comparing mid-level features for transient attribute estimation. (left) Features extracted from the weights of our proposed approach versus a network trained for image classification and a randomly initialized baseline. (right) Features extracted from our method, trained on different datasets.

we compare to features of the same architecture trained for ImageNet [17] classification and features sampled uniformly at random. We select the feature from the last pooling layer (*AvgPool\_1a\_7x7*), which is the deepest layer that this model and ours share in common. We compute the cross correlation scores between the feature and the transient attribute scores of each image, resulting in a  $1024 \times 40$  cross correlation matrix,  $M$ , where the element  $m_{ij}$  is the cross correlation score between the  $i$ -th feature channel and the  $j$ -th transient attribute. Figure 4 shows, for each transient attribute, the maximum absolute correlation score over all feature channels. We observe that our proposed method learns features that are more correlated to the transient attributes ( $\bar{\rho} = 0.414$ ) than the ImageNet network ( $\bar{\rho} = 0.281$ ) or the random features ( $\bar{\rho} = 0.038$ ).

#### 4.4 Comparing Mid-Level Features for Transient Attribute Estimation

The previous experiment showed that the image context network is capturing mid-level features correlated with transient attributes. In this section, we explore the ability of this representation for directly estimating transient attributes. Similar to the previous experiment, we truncate our model at the last pooling layer (in order to compare versus alternative initialization strategies), and add a final two-layer MLP with 40 outputs corresponding to the 40 transient attributes in the *TA* dataset. We train this network, initializing from the weights of models trained for different tasks, including variants of our method trained on the *AMOS*, *YFCC*, and *Hybrid* datasets. During training, the MLP portions are randomly initialized while the earlier layers are frozen. We evaluate the average mean squared error (MSE) for the test set every 500 iterations (batch size 32). Figure 5 shows the performance comparison among different mid-level features, including ImageNet and randomly initialized *InceptionV2*. Our features are superior to all baselines and perform best when learned using the *Hybrid* dataset.

#### 4.5 Application: Image Localization

There are two image localization formulations that our network architecture enables. The straightforward approach is to use the location estimator (or the time-conditioned variant) to generate a probability distribution over a discrete set of location bins. An alternative

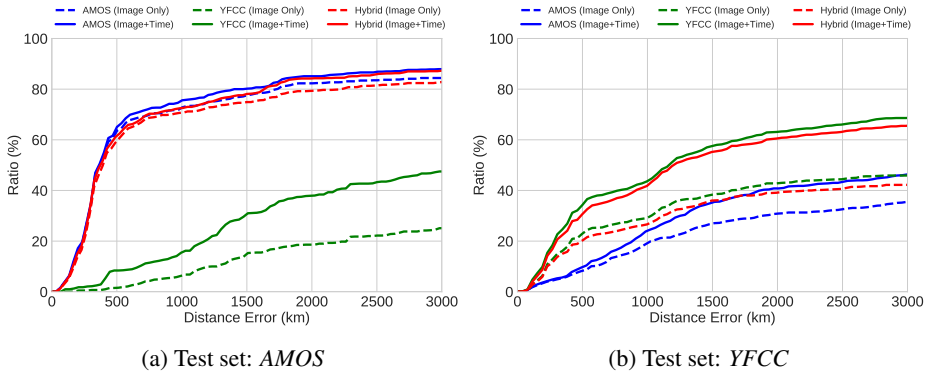


Figure 6: Quantitative evaluation of localization performance shown as a cumulative distance error plot for the *YFCC* dataset.

approach is to optimize for a continuous location estimate by minimizing the loss of the location-conditioned time estimator.

**Discrete Localization** Given an input image,  $I$ , we evaluate the location estimator  $P(I|C_I(I))$  and the time-conditioned location estimator  $P(I|C_I(I), C_T(t))$ , which requires a timestamp,  $t$ . We trained our model on each dataset and perform quantitative evaluation using the test images from *AMOS* and *YFCC*, separately. We use the latitude/longitude center of the highest probability bin as our location estimate. The results of this experiment are presented in Figure 6. We observe that the time-conditioned location estimator is superior in both cases. We also conclude that our model performs better if trained on the same imagery source with the test set, and training the network with the *Hybrid* dataset is competitive on both test sets.

**Continuous Localization** In this formulation, we use the location-conditioned time estimator,  $P(t|C_I(I), C_L(I))$ , to optimize for a continuous location estimate. Given the known image capture time  $t^*$ , the idea is that the true location should result in a low value for the loss associated with the estimator,  $\ell_t = \phi(P(t|C_I(I), C_L(I)), t^*)$ , where  $\phi(\circ, \circ)$  is the cross-entropy loss. Therefore, we can produce a location estimate by optimizing the location,  $I$ , with respect to  $\ell_t$ . Unfortunately, an individual image does not typically yield a unique, or accurate, location estimate using this method. However, if we sum the loss across images captured at different times, we find that the minima of the function becomes more distinct. Figure 7 shows several qualitative examples of this localization strategy on static webcams, where darker colors correspond to more likely locations. We can see that as additional images are included in the loss, the uncertainty of the location prediction diminishes.

## 4.6 Application: Time Estimation

Using the time estimator and location-conditioned time estimator, our network is able to estimate the capture time of a query image. These estimators output a distribution in discrete 2D time space. To evaluate our estimates, we compare the ground-truth capture time and the marginal probabilities of our predictions on the *YFCC* test set, and present the cumulative error plots in Figure 8. We observe that including location is not useful for pinpointing the

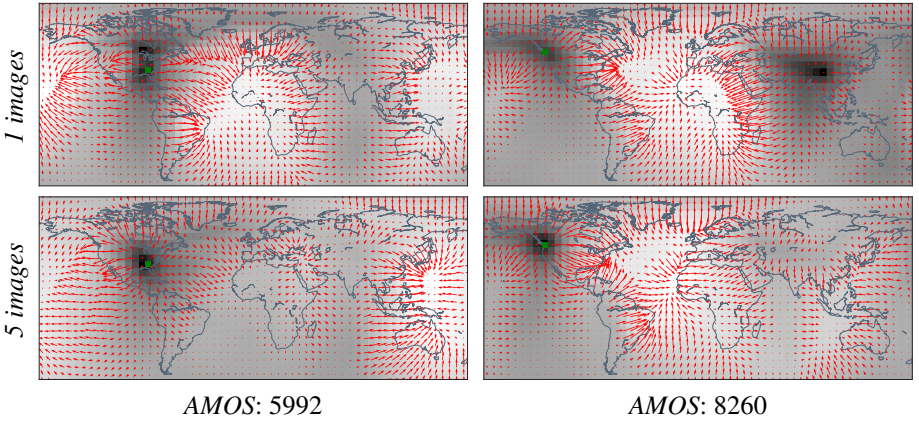


Figure 7: Visualizing the time estimation loss for two webcams and varying number of images (darker is lower). The red arrows show the gradient and the green dot is the true location.

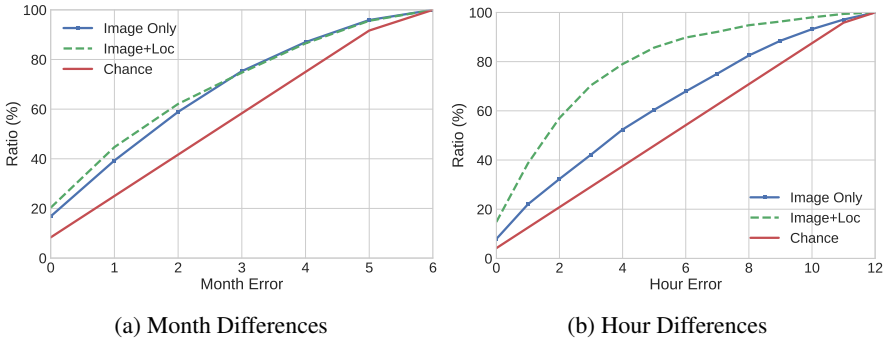


Figure 8: Quantitative evaluation of marginal time estimation performance, shown as cumulative error plots for the *YFCC* dataset. Both methods perform better than the random chance and including the known location results in a significant reduction in error for the hour-estimation task.

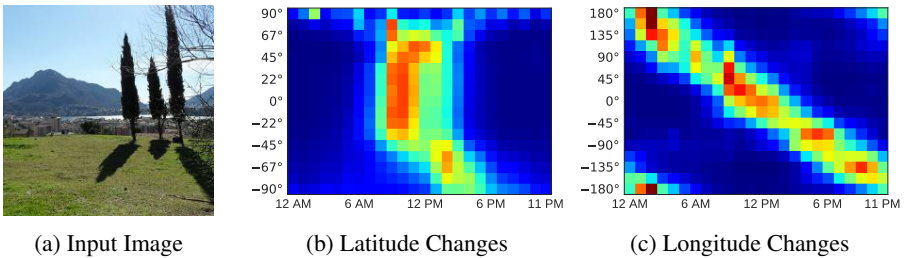


Figure 9: When using our location-conditioned time estimator, the marginal probability over hours changes significantly as we vary the latitude and longitude provided to the location context network.

month. We suspect this is because most of our imagery is in the northern hemisphere, and changing the location within a hemisphere doesn't change the season. However, this is not the case when estimating the hour. To visualize this, we show in Figure 9 the impact of changing the location on the hour estimate. We compute the marginal hour distribution at different latitudes and longitudes. When performing a sweep over latitude, we fix the longitude value to be the ground truth (and vice versa). We found that the longitude of the image corresponds more with the hour prediction than the latitude, which matches expectations.

## 5 Conclusion

When learning about the world using images, the location and time an image was captured are useful pieces of metadata that are often available, but commonly overlooked. We presented a novel architecture for learning useful representations from images that takes advantage of this metadata. We found that for the task of transient attribute estimation, our method, despite being trained without manually obtained image-level annotations, learned image representations that outperform the representations learned using ImageNet. This is a rarely achieved feat in self-supervised representation learning against a frequently used baseline. One important area for future work is in investigating alternative architectures for the context networks. We did not conduct a thorough study in this regard and expect to see improvements in using newer image CNNs and higher capacity time and location networks. In addition, we expect that richer time and location input representations will result in improved geo-temporal image features.

## Acknowledgement

We gratefully acknowledge the support of NSF CAREER award IIS-1553116 and ARPA-E Award DE-AR0000594. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

## References

- [1] Ryan Baltenberger, Menghua Zhai, Connor Greenwell, Scott Workman, and Nathan Jacobs. A Fast Method for Estimating Transient Scene Attributes. In *IEEE Winter Conference on Applications of Computer Vision*, 2016.
- [2] Carl Doersch. *Supervision Beyond Manual Annotations for Learning Visual Representations*. PhD thesis, Carnegie Mellon University, 2016.
- [3] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *IEEE International Conference on Computer Vision*, pages 1422–1430, 2015.
- [4] Roman Fedorov, Piero Fraternali, and Marco Tagliasacchi. Snow phenomena modeling through online public media. In *IEEE International Conference on Image Processing*, 2014.
- [5] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

- [6] James Hays and Alexei A. Efros. im2gps: estimating geographic information from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15*, pages 448–456. JMLR.org, 2015.
- [8] Mohammad T. Islam, Nathan Jacobs, Hui Wu, and Richard Souvenir. Images+weather: Collection, validation, and refinement. In *IEEE CVPR Workshop on Ground Truth*, 2013.
- [9] Nathan Jacobs, Nathaniel Roman, and Robert Pless. Consistent temporal variations in many outdoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
- [10] Nathan Jacobs, Scott Satkin, Nathaniel Roman, Richard Speyer, and Robert Pless. Geolocating static cameras. In *IEEE International Conference on Computer Vision*, 2007.
- [11] Nathan Jacobs, Mohammad T. Islam, and Scott Workman. Cloud motion as a calibration cue. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
- [12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [13] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics (SIGGRAPH)*, 33(4):149, 2014.
- [14] Jean-François Lalonde, Srinivasa G Narasimhan, and Alexei A Efros. What do the sun and the sky tell us about the camera? *International Journal of Computer Vision*, 88(1):24–51, 2010.
- [15] Tsung-Yi Lin, Yin Cui, Serge Belongie, and James Hays. Learning deep representations for ground-to-aerial geolocalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.
- [16] Cewu Lu, Di Lin, Jiaya Jia, and Chi-Keung Tang. Two-class weather classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
- [17] Kevin Matzen and Noah Snavely. Scene chronology. In *European Conference on Computer Vision*, 2014.
- [18] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [19] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [20] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.
- [21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.



- [22] Tawfiq Salem, Scott Workman, Menghua Zhai, and Nathan Jacobs. Analyzing Human Appearance as a Cue for Dating Images. In *IEEE Winter Conference on Applications of Computer Vision*, 2016.
- [23] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [24] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data and new challenges in multimedia research. *CoRR*, abs/1503.01817, 2015.
- [25] Anna Volokitin, Radu Timofte, and Luc Van Gool. Deep features or not: Temperature and time prediction in outdoor scenes. In *CVPR Workshop on Robust Features*, 2016.
- [26] Jingya Wang, Mohammed Korayem, and David J Crandall. Observing the natural world with flickr. In *IEEE International Conference on Computer Vision Workshops*, 2013.
- [27] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. In *European Conference on Computer Vision*, pages 37–55. Springer, 2016.
- [28] Scott Workman and Nathan Jacobs. On the location dependence of convolutional neural network features. In *IEEE/ISPRS Workshop: Looking From Above: When Earth Observation Meets Vision*, 2015.
- [29] Scott Workman, R. Paul Mihail, and Nathan Jacobs. A pot of gold: Rainbows as a calibration cue. In *European Conference on Computer Vision*, 2014.
- [30] Scott Workman, Richard Souvenir, and Nathan Jacobs. Wide-area image geolocalization with aerial reference imagery. In *IEEE International Conference on Computer Vision*, pages 1–9, 2015.
- [31] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision*, 2016.
- [32] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.