BubbLeNet Foveated imaging for visual discovery

{kmatzen,snavely}@cs.cornell.edu http://bit.ly/bubblenets

Kevin Matzen
Noah Snavely

Cornell University

Problem Statement

Given a corpus of categorized photos, we aim to produce a concise visual representation of what makes each category distinctive. In turn, we use this representation in order to make **measurements** across Internet-scale datasets to learn about stylistic variations of people and places across space and time.

Why do we desire a concise, visual representation?
A visual representation is useful since it can be presented to humans to give them insight regarding trends within the dataset. A concise representation is preferred to minimize the effort the analyst must expend in order to understand the results.

Therefore, we revisited the **discriminative patches** problem. We addressed this problem with two questions in mind:

(1) Convolutional neural networks (CNNs) have internal representations that outperform hand-engineered features on many classification tasks. These hand-engineered features form the basis for most prior discriminative patches work. **How can we incorporate CNN features cleanly in a discriminative patches framework?**

(2) Billions of photos are uploaded per day to services such as Facebook and Instagram. It would be useful if we could measure trends relating to elements represented by these sets of patches across both space and time. **How should we design the representation such that this detection step is inexpensive?**

Main Idea

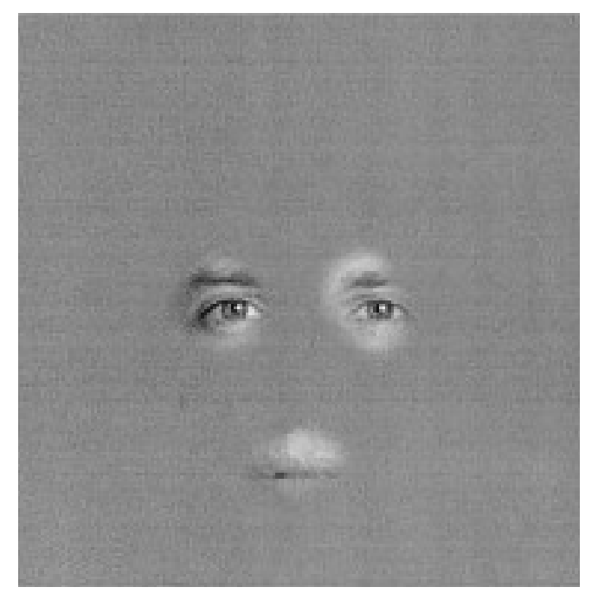
Adapt prior method for studying human perception to find discriminative regions.

[1] presents a method for determining what makes humans perceive differences between attributes such as gender and facial expression. We construct analogous experiments with **CNNs** and **Internet-scale datasets**.

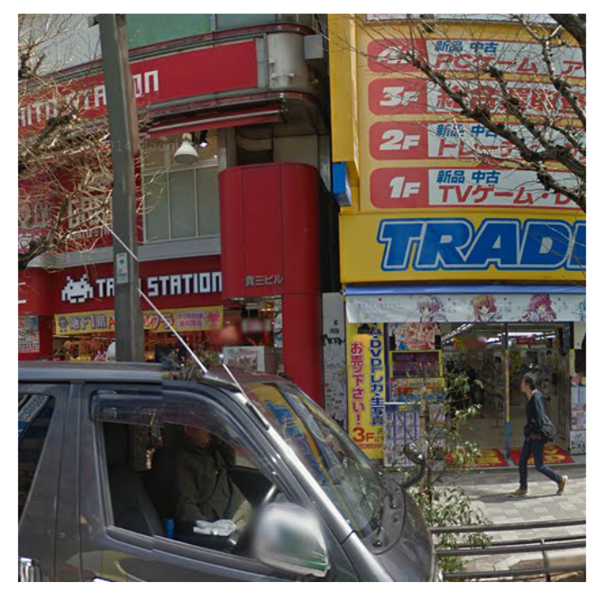
Previous experiment

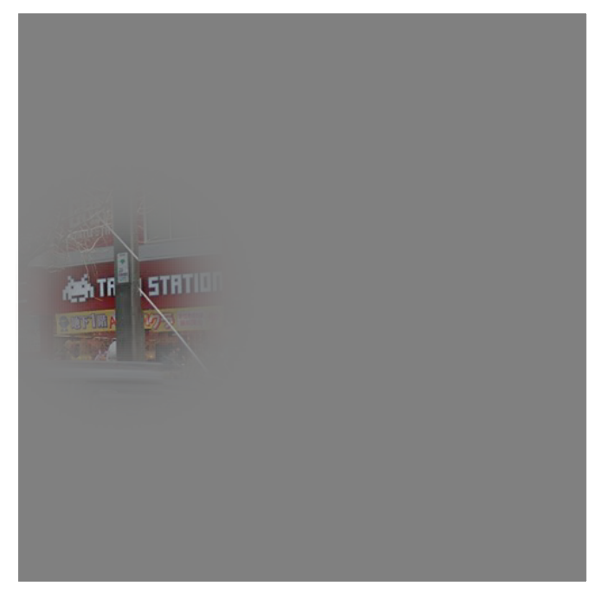
What regions make humans perceive this person as "male"?

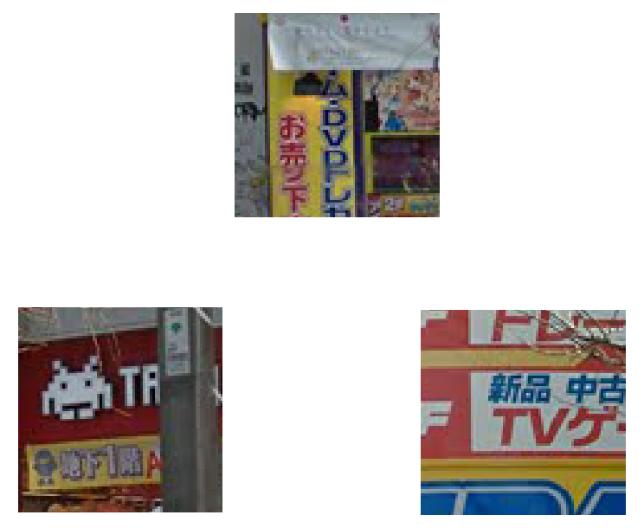
Ask human subjects to classify masked images.


Identify discriminative regions based on successful classifications.

Our adaptation


What regions make **CNNs** perceive this location as **"Tokyo"**?


Ask **CNNs** to classify masked images.


Identify discriminative **patches** based on successful classifications.

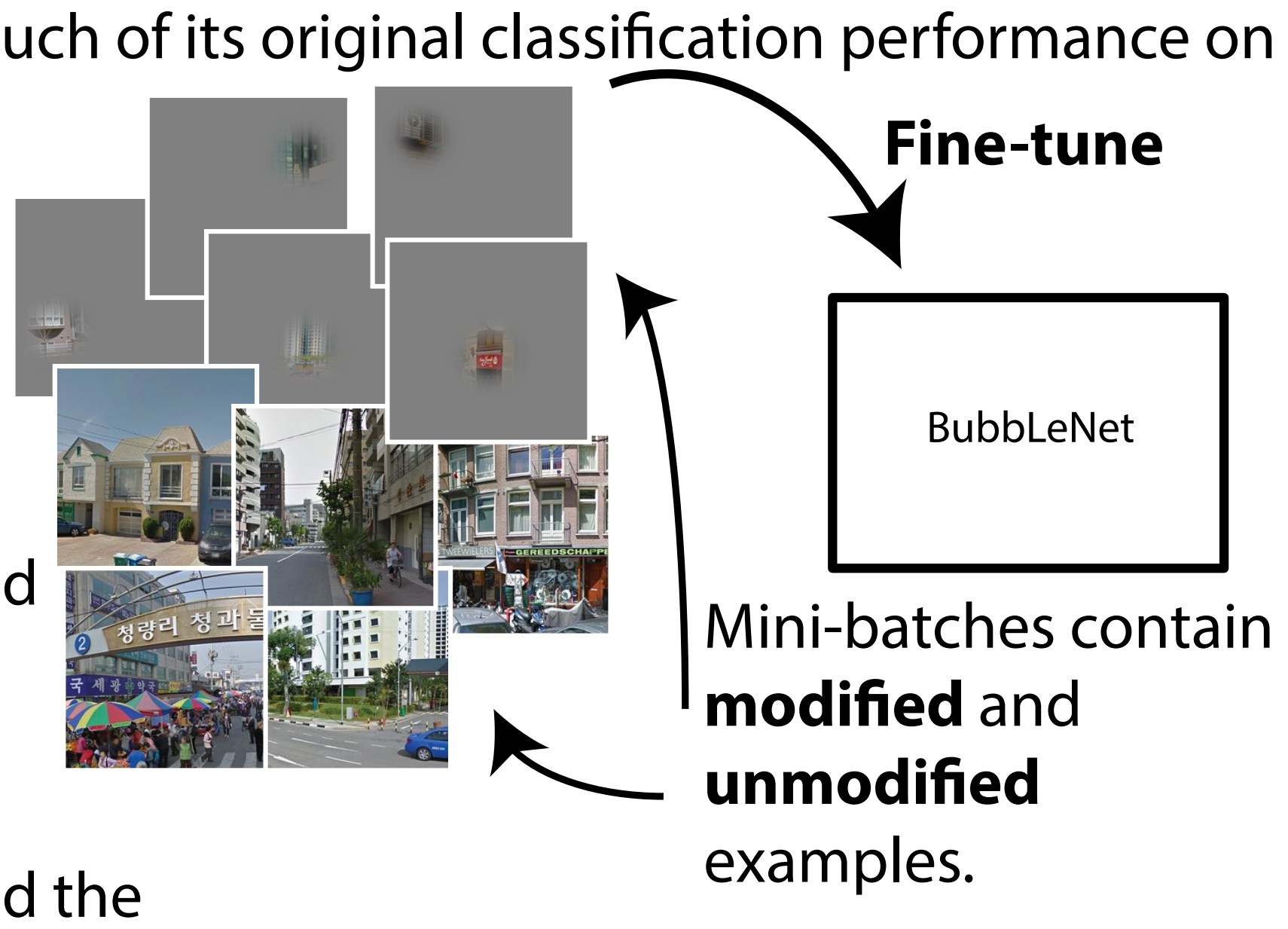
Key Observation: CNNs that perform well on the original classification task tend not to perform well on the masked image task. Can we train CNNs to make reasonable predictions with such a tiny amount of the image visible?

Method

Our method has four steps.

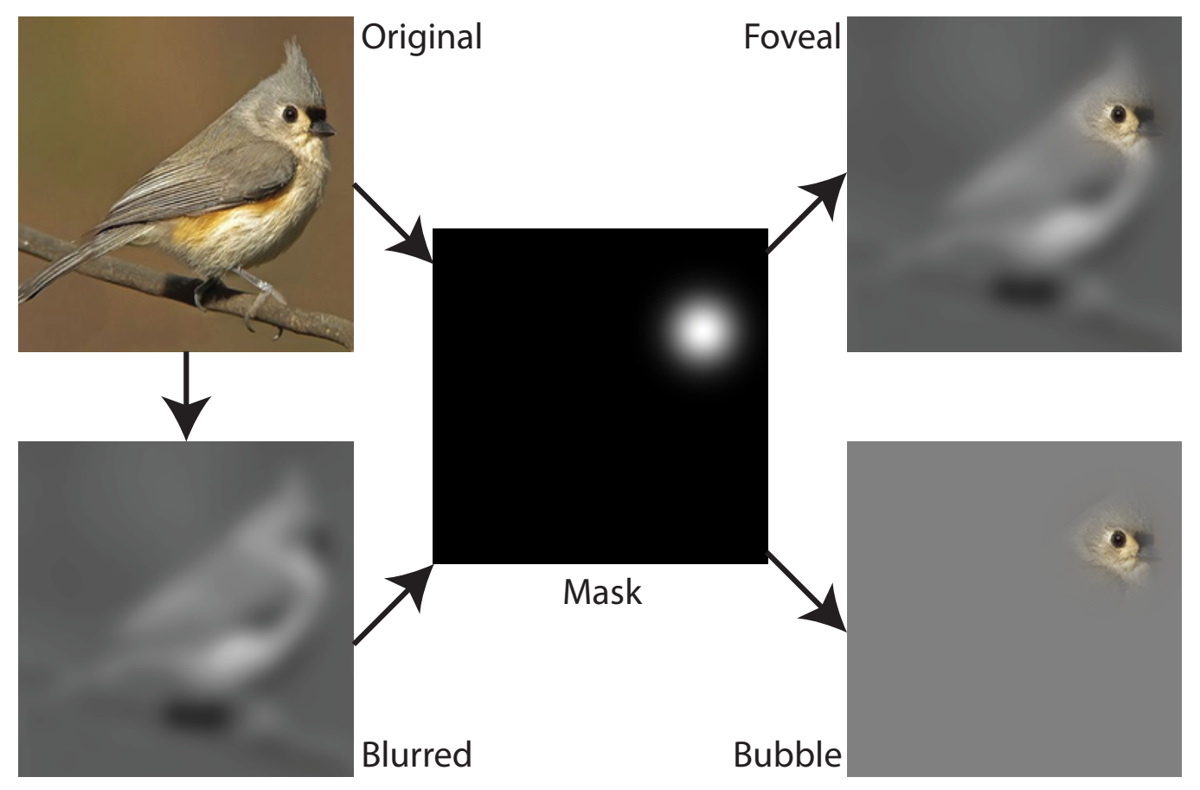
(1) **Train a standard CNN for the target task.**
We apply our method to a number of datasets such as ILSVRC2012, Places205, and CUB-200-2011. If a trained network is available, we use that as the base network. If not, then we fine-tune an existing network.

(2) **Continue to fine-tune the classifier with a mix of modified and unmodified images.**
We wish to arrive at a network that can achieve non-trivial performance given modified images, but has retained much of its original classification performance on unmodified images.


Fine-tune

Mini-batches contain **modified** and **unmodified** examples.

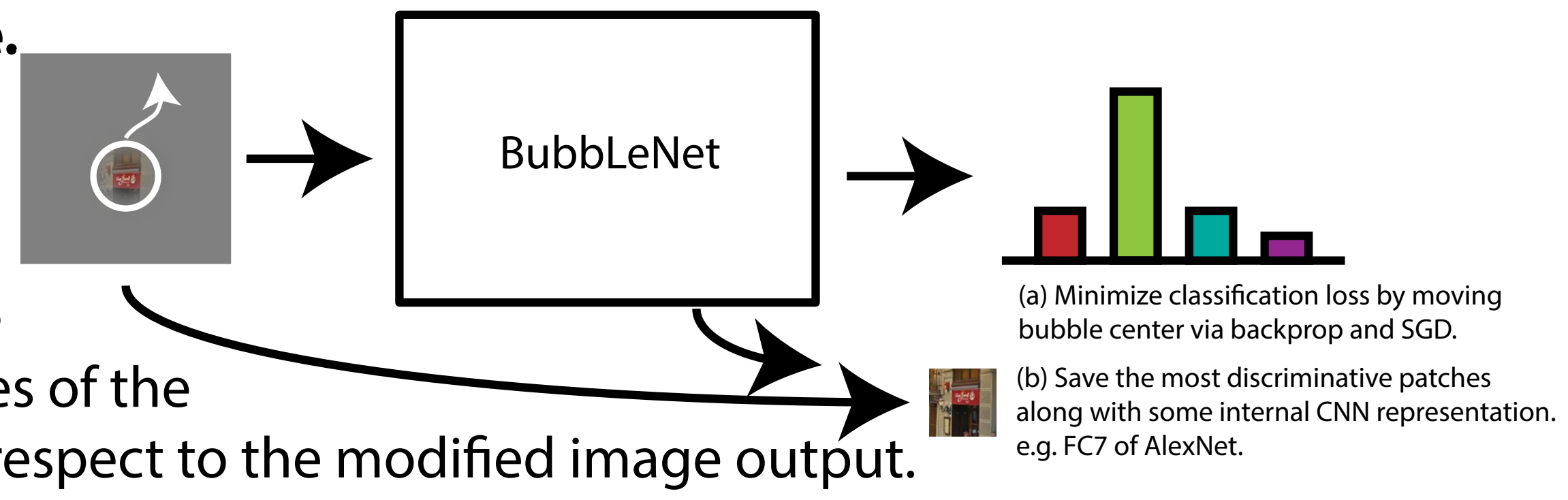
We tried two types of modification to produce two types of networks: **FoveaNets** and **BubbLeNets**.
FoveaNet: Modified images have a bubble of unobscured content and the rest is heavily blurred. (Inspired by [2])
BubbLeNet: Modified images have a bubble of unobscured content and the rest is completely deleted. (Inspired by [1])



Both representations were trainable, so we favor the **BubbLeNet** since it is easier to reason about what content outside the bubble is leaked. e.g. Foveal images can leak shape information.

During training, bubble locations were sampled at random and mini-batches contained 50% modified images.

(3) **Use supervised objective function + backprop to select one discriminative patch per image.**



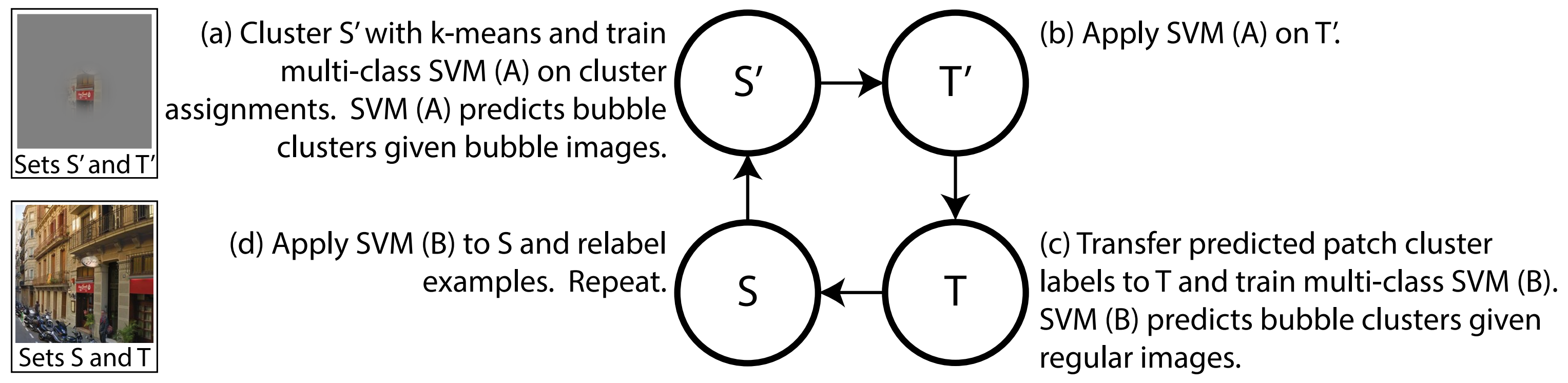
(a) Minimize classification loss by moving bubble center via backprop and SGD.

(b) Save the most discriminative patches along with some internal CNN representation. e.g. FC7 of AlexNet.

Both FoveaNet and BubbLeNet representations allow us to compute the partial derivatives of the bubble center with respect to the modified image output. This means we can add this pre-processing operation as a layer in our network. This offers an efficient alternative to densely evaluating bubble positions.

(4) **Cluster patches and rank clusters.**
Inspired by [3], we build a set of patch clusters as well as a ranking function to help promote the most important clusters.

We split our validation set of unmodified images into two partitions — S and T. Corresponding sets S′ and T′ contain their discriminative patch counterparts.



(a) Cluster S′ with k-means and train multi-class SVM (A) on cluster assignments. SVM (A) predicts bubble clusters given bubble images.

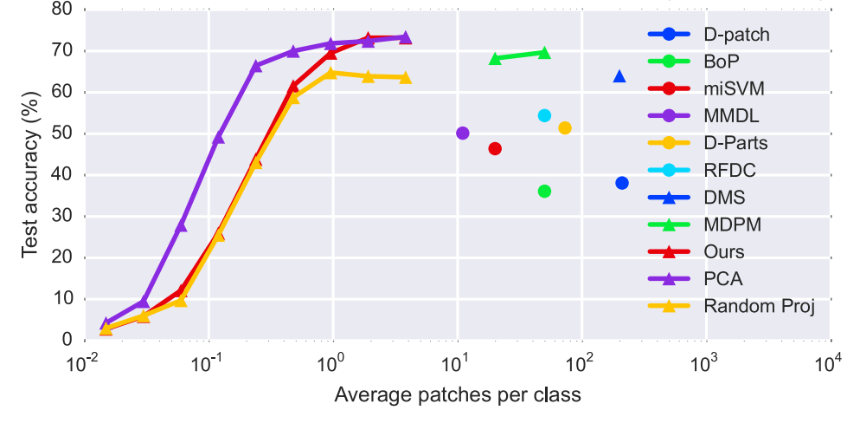
(b) Apply SVM (A) on T′.

(c) Transfer predicted patch cluster labels to T and train multi-class SVM (B). SVM (B) predicts bubble clusters given regular images.

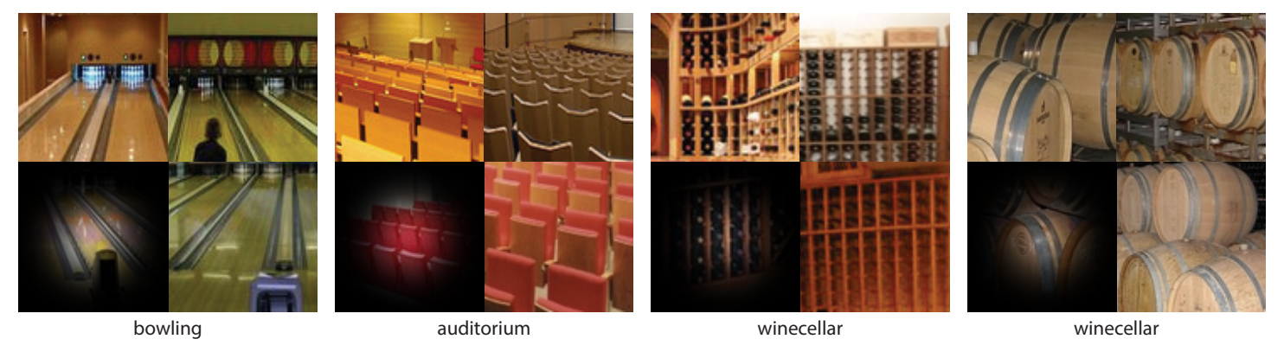
(d) Apply SVM (B) to S and relabel examples. Repeat.

Validation

How **representative** are these patch clusters? i.e. Do a few clusters capture the discriminative power of the original classifier?
We applied an algorithm similar to BubbleBank [2] to the Indoor67 dataset. While prior work allocates tens of elements to each class, we found that on average one class could be represented by only two of our clusters.

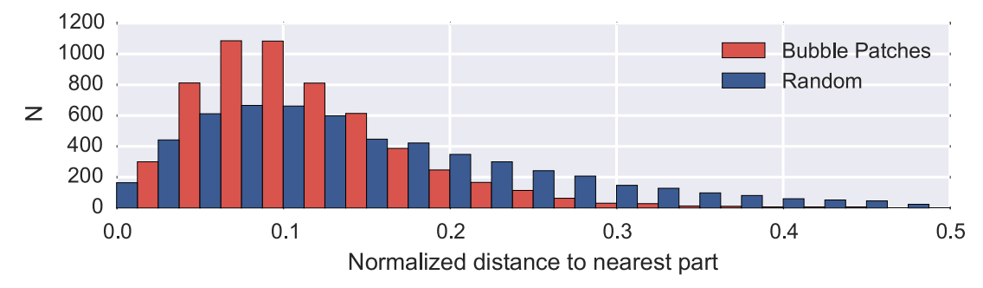


Indoor67 classification performance.

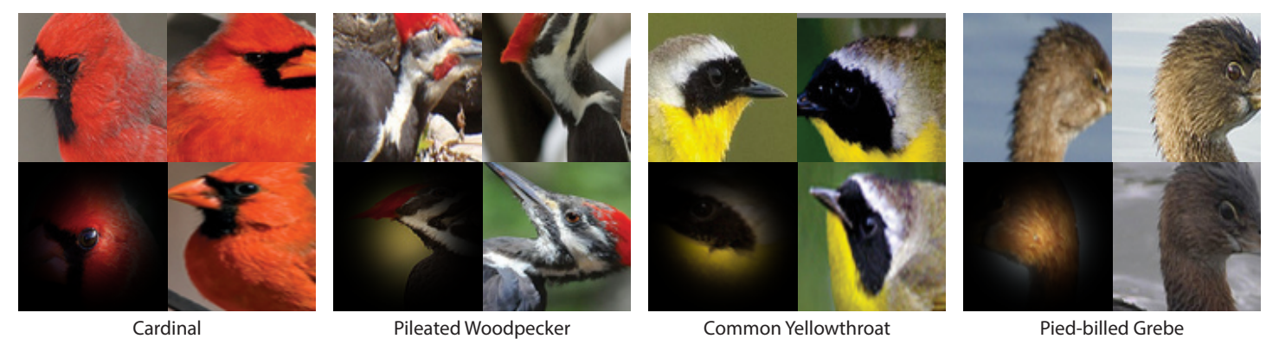


Example Indoor67 clusters. An example with mask is shown for scale.

How well do bubbles **localize** discriminative parts?
We applied our method to CUB-200, a dataset of birds with part annotations. We found that our method was much more likely to choose a bubble location near a part annotation compared to chance.



Histogram of distance from mined patch to closest part. Image coordinates have been normalized to a 1x1 square.

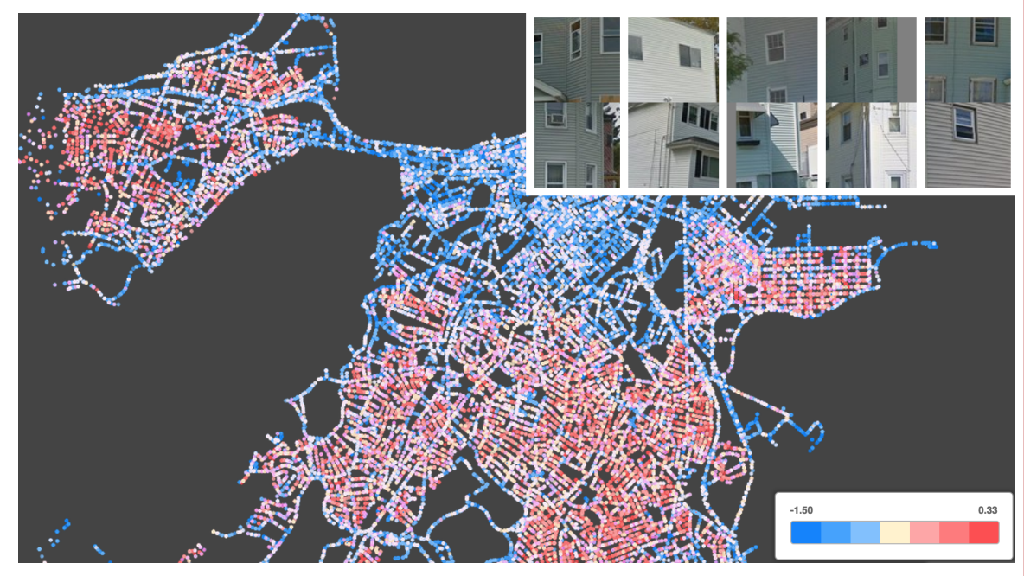


Example CUB-200-2011 clusters. An example with mask is shown for scale.

Applications

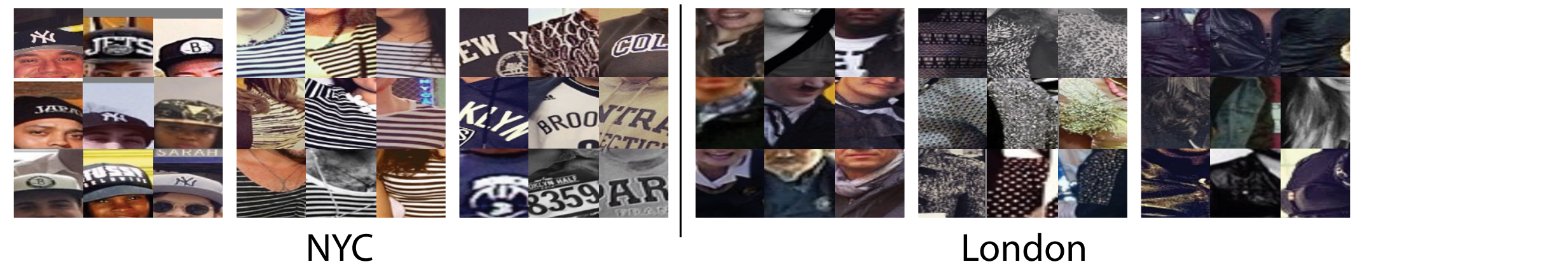
We applied our method to two Internet-scale datasets of **people** (Instagram) and **places** (Street View) to discover discriminative visual elements and to measure their occurrence across space and time.

Street View for spatial analysis
A BubbLeNet trained on city classification revealed that architectural elements and street markers play an important role in city classification (Inspired by [4]). Here we show that an important visual element for Boston is siding on homes. We applied the detector for this visual element to a much larger set of photos to build a spatial visualization.



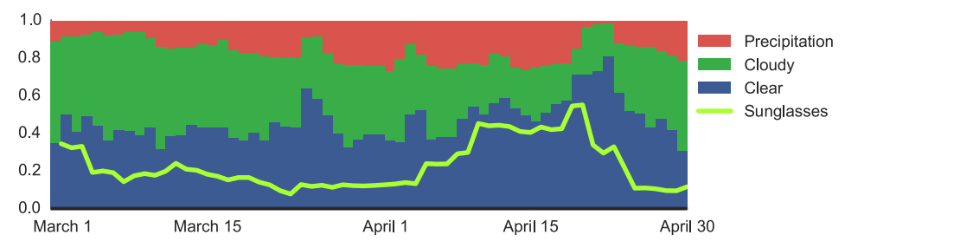
Example patches in a Boston-centric cluster (inset). Activations of this cluster spanning Boston. Red (blue) high (low) activation.

Instagram for spatial analysis
Given cropped detections of people in Instagram photos, we trained a BubbLeNet to predict whether the photo was taken in NYC or London (60.8% accuracy). Top ranked clusters include different types of clothing , logos, and fabric patterns.


NYC London

Instagram for temporal analysis
We also trained a classifier to predict the month given a photo of a person in NYC. Certain months were more distinguishable than others. For example, the top ranked cluster for March versus April was "sunglasses" and we used this detector to measure the occurrence of sunglasses over time. We can corroborate this finding using weather data.

Example patches from the "sunglasses" cluster.



"Sunglasses" activations over time. Weather data from wunderground.com. Sunglasses-Clear r = 0.487. Sunglasses-Cloudy r = -0.593.

References
[1] Gosselin and Schyns, "Bubbles: a technique to reveal the use of information in recognition tasks", Vision Research 2001.
[2] Deng, et. al., "Fine-Grained Crowdsourcing for Fine-Grained Recognition", CVPR 2013.
[3] Singh, et. al., "Unsupervised discovery of mid-level discriminative patches", ECCV 2012.
[4] Doersch, et. al., "What makes Paris look like Paris?", SIGGRAPH 2012.

Acknowledgments
This work is partly funded by NSF grant IIS-1111534, Google, and Intel ISTC-VC.