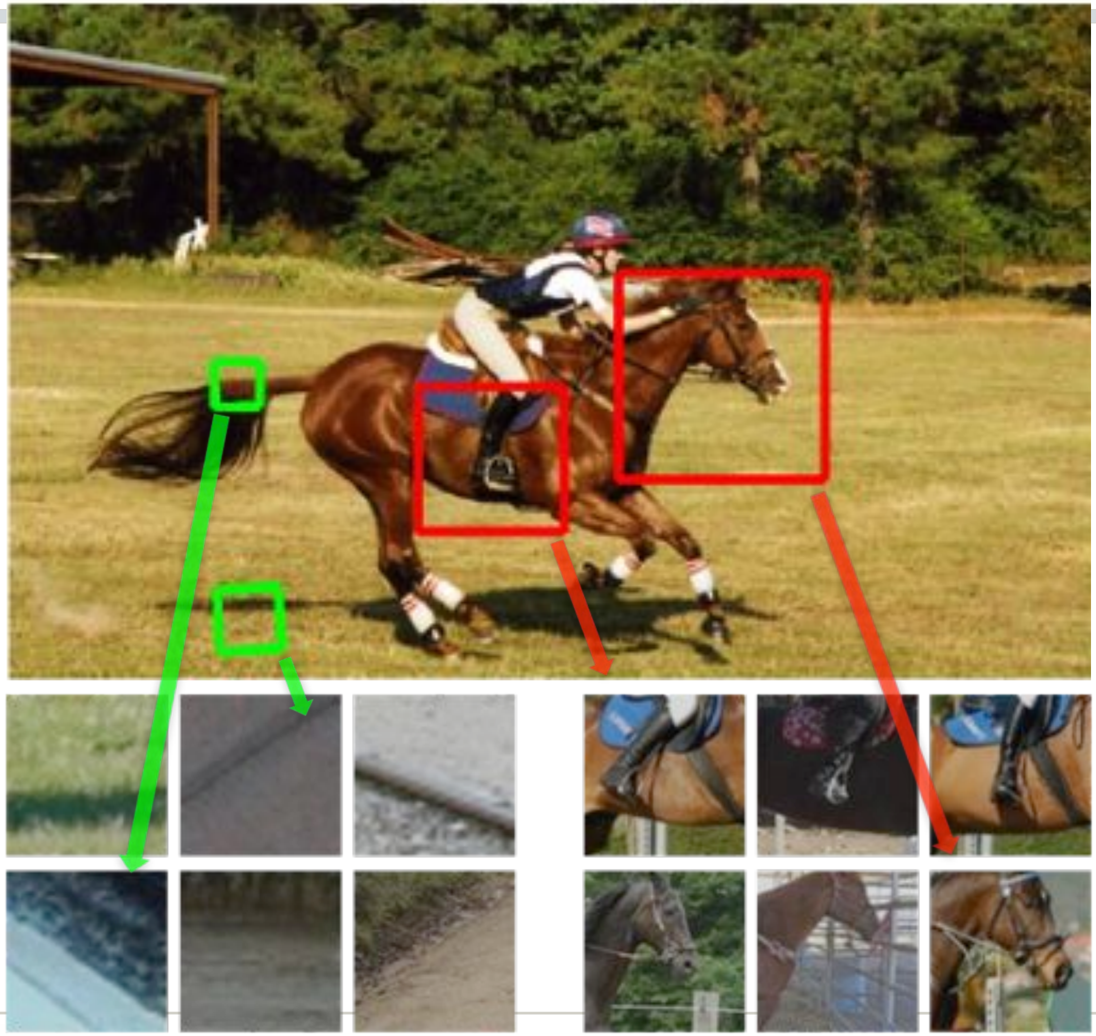
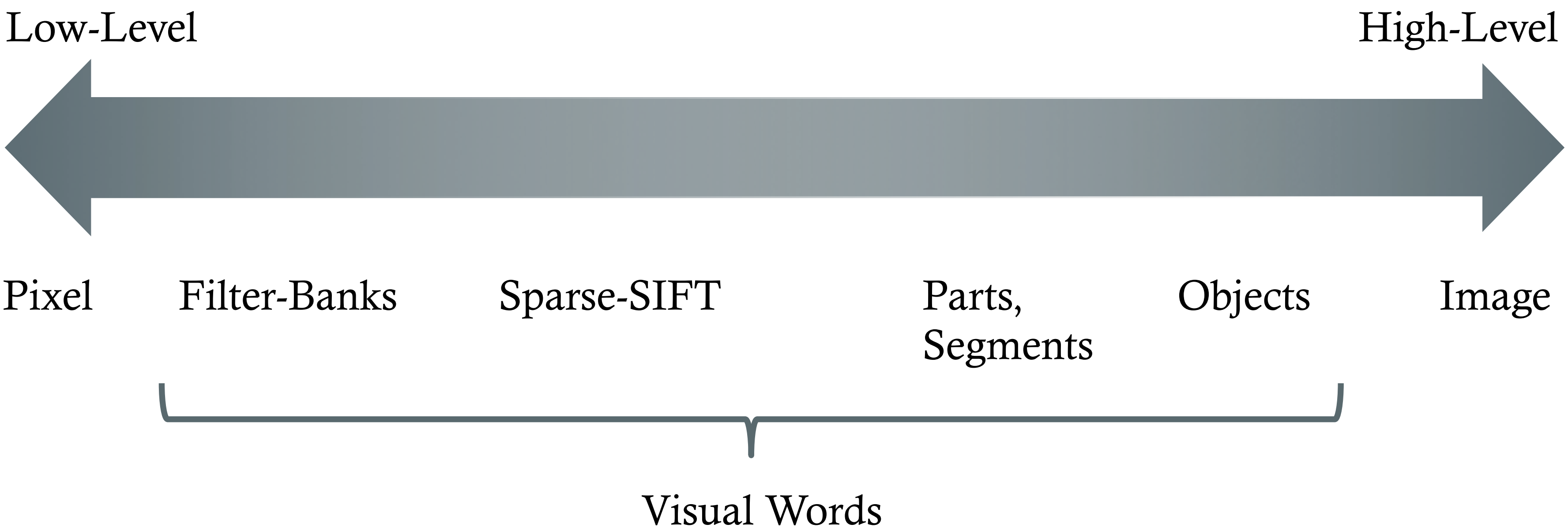
Unsupervised Discovery Of Mid-level Discriminative Patches

Saurabh Singh (ss1@andrew.cmu.edu), RI

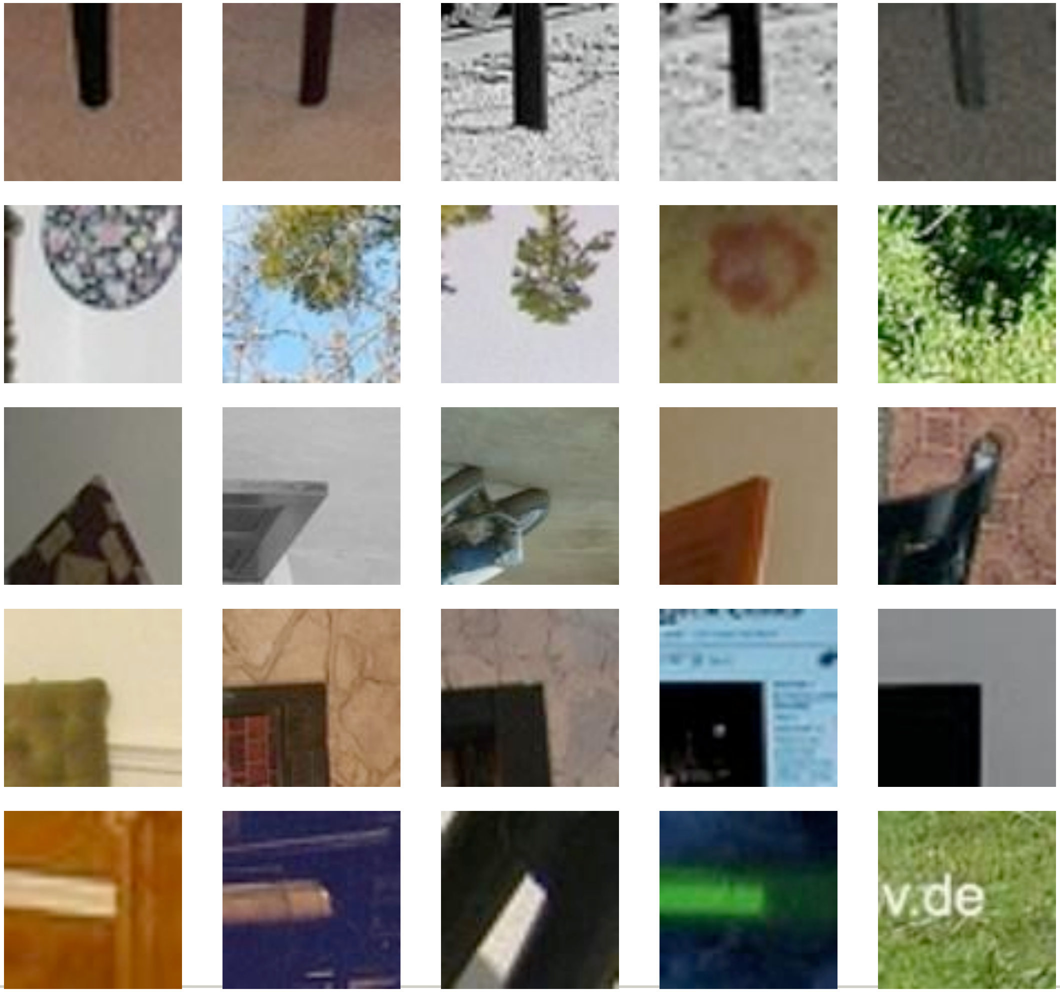
Which representation seems intuitive?



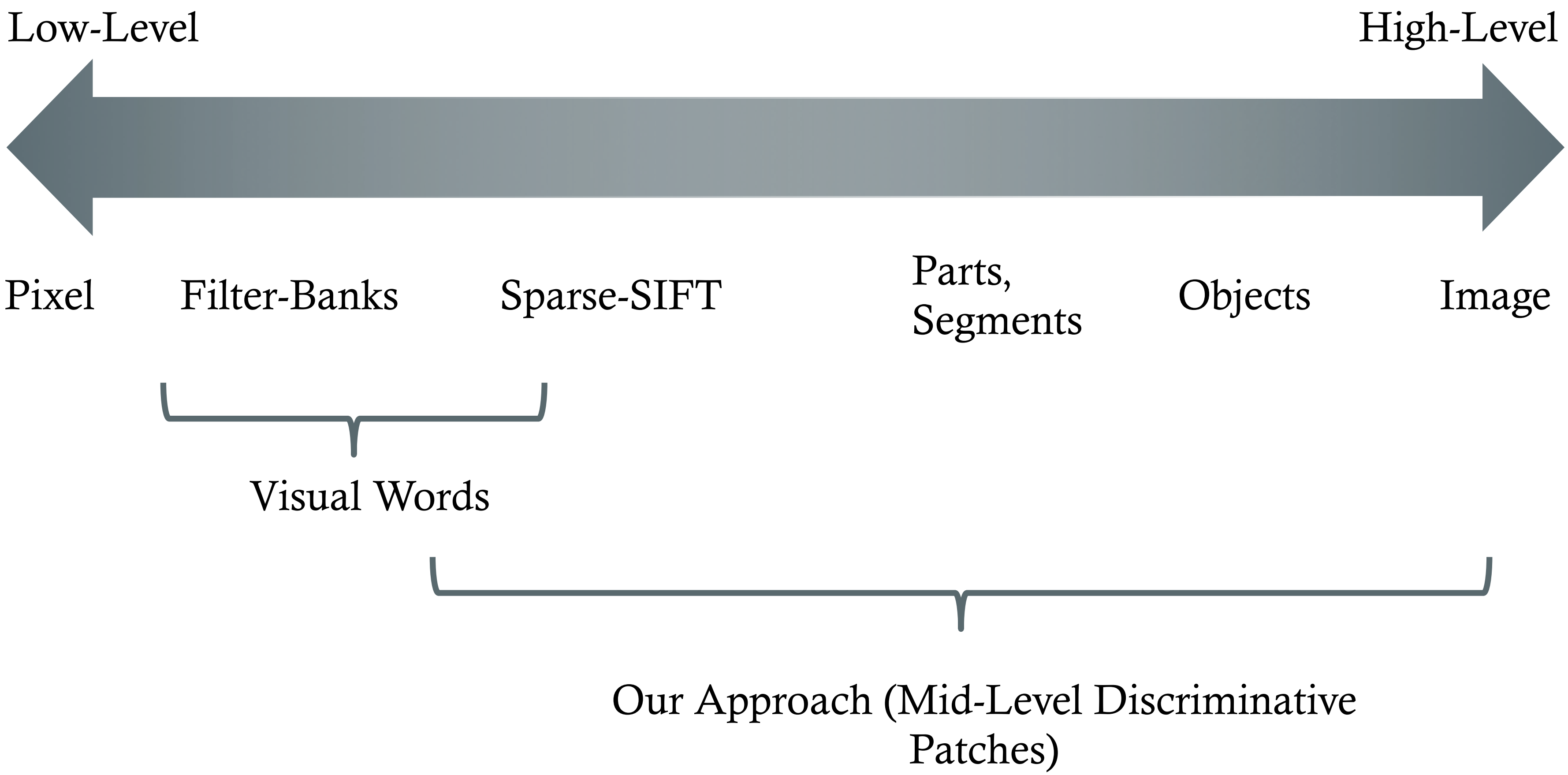
Spectrum of Visual Features



Visual Words or Letters?



Spectrum of Visual Features

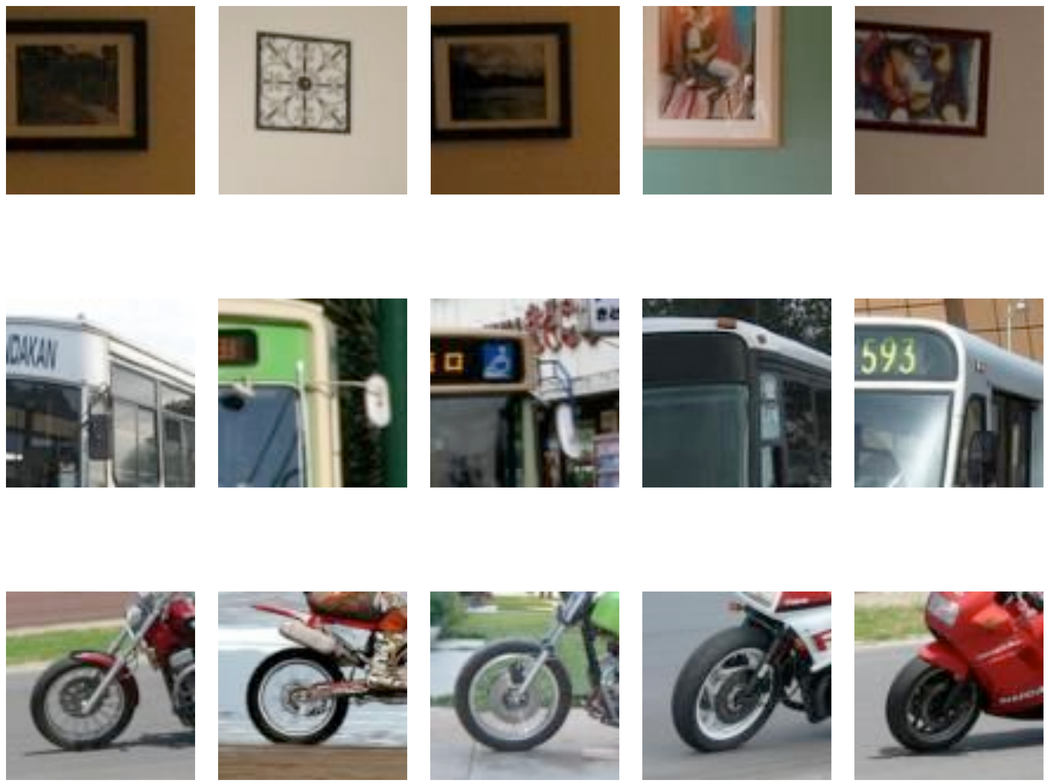


Discriminative Patches

Two key requirements

1. Representative : Need to occur frequently enough.
2. Discriminative: Need to be different enough from the rest of the visual world.

First some examples



Unsupervised Discovery of Discriminative Patches

Given “discovery dataset”

Find a relatively small number of discriminative patches that represent it well.

We assume access to a “natural world” dataset, which captures the visual statistics of the world in general.

Dataset: Subset of Pascal VOC 2007 with six categories.

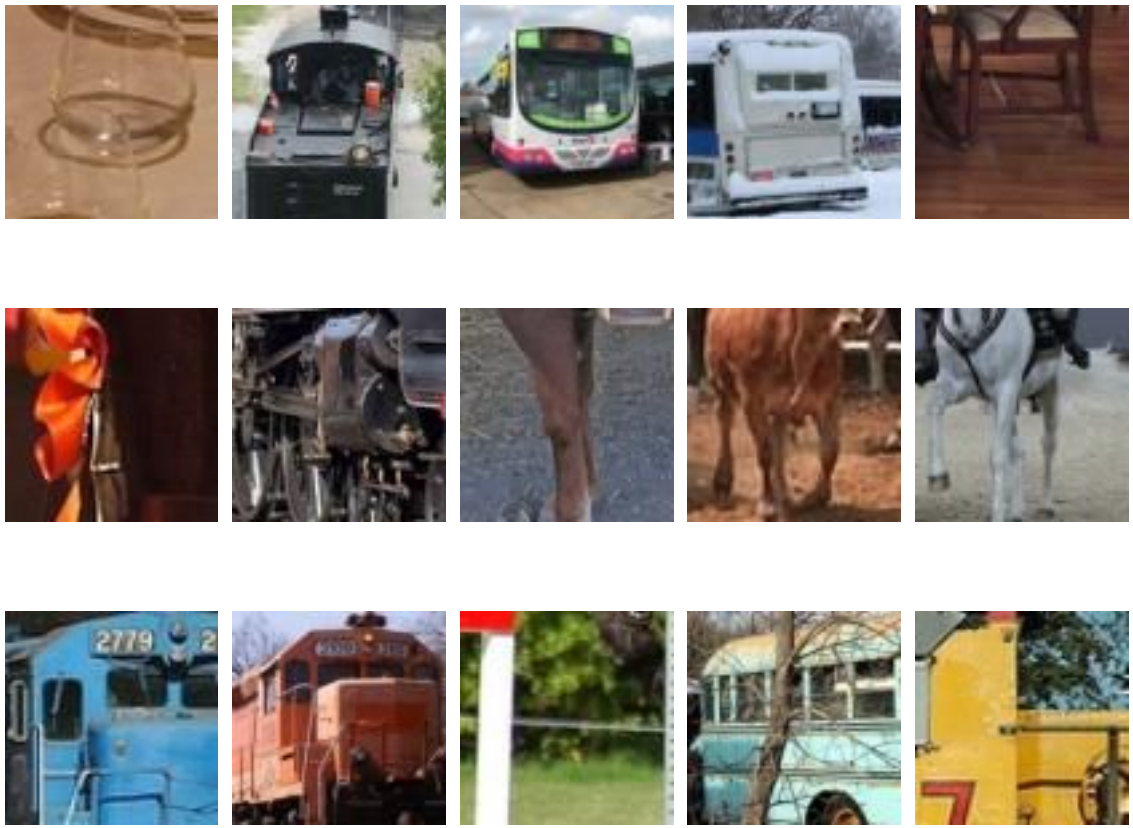
Visual Word Approach

- Sample a lot of patches from the discovery dataset (represented in terms of their features*) at various locations and scales.
- Perform some form of unsupervised clustering (e.g. K-Means)

Doesn't work well.

* We use Histogram of Oriented Gradients (HOG) features

K-Means Clusters



Chicken-Egg Problem

- If we know that a set of patches are visually similar, we can easily learn a distance metric for them
- If we know the distance metric, then we can easily find other members.

Discriminative Clustering

- Initialize using K-Means
- Train a discriminative classifier to represent the distance function (treating other clusters as negative examples).
- Re-assign the patches to clusters whose classifier gives highest score
- Repeat

Discriminative Clustering*

- Initialize using K-Means
- Train a discriminative classifier to represent the distance function (Using “natural world” as negative data).
- Detect the patches and assign to clusters.
- Repeat

Discriminative Clustering*

Initial

Final

Initial

Final

Discriminative Clustering+

- Split the discovery dataset into two equal parts {Training, Validation}
- Perform the training step of Discriminative Clustering* on Training set.
- Perform the detection step of Discriminative Clustering* on Validation set.
- Exchange the roles of Training and Validation sets.
- Repeat.

Discriminative Clustering+

KMeans

Iter 1

Iter 2

Iter 3

Iter 4

Discriminative Clustering+

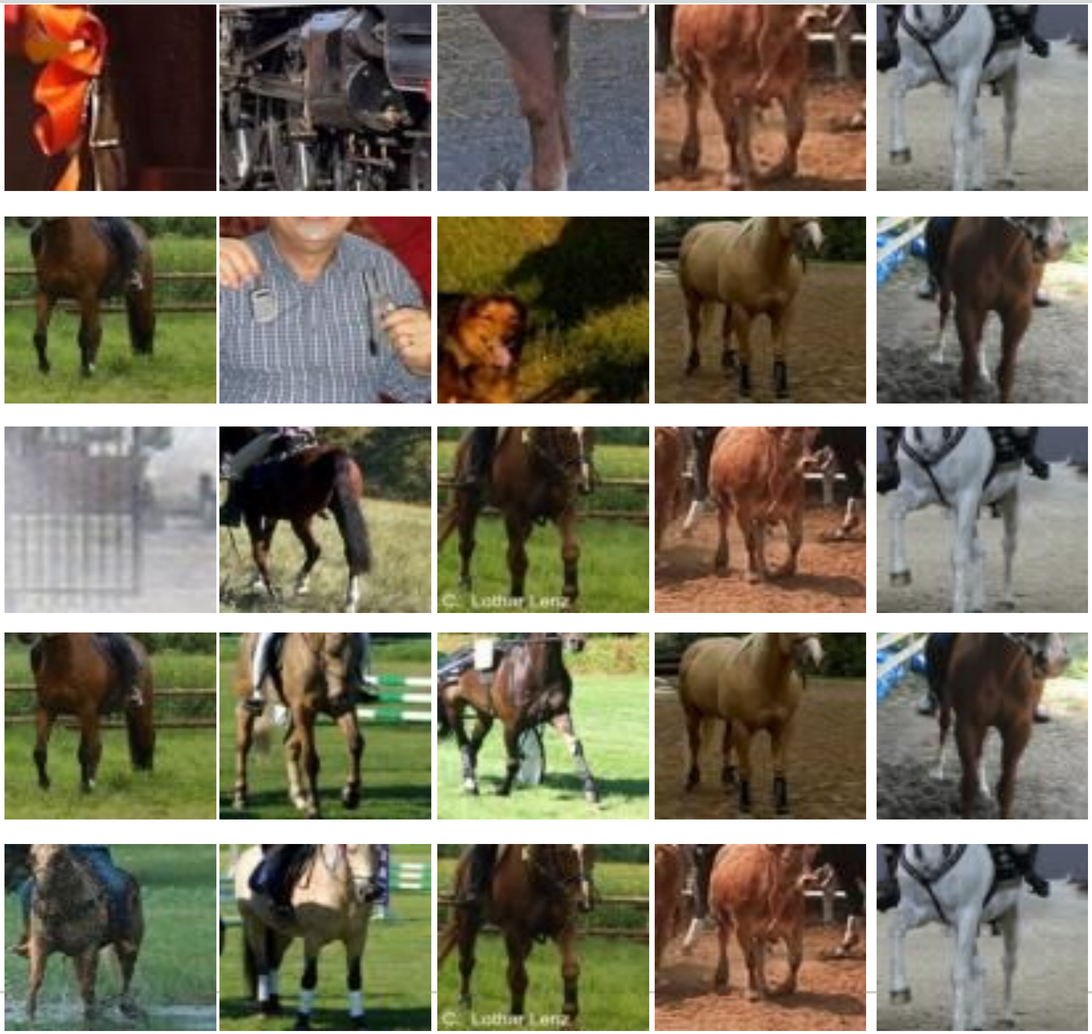
KMeans

Iter 1

Iter 2

Iter 3

Iter 4



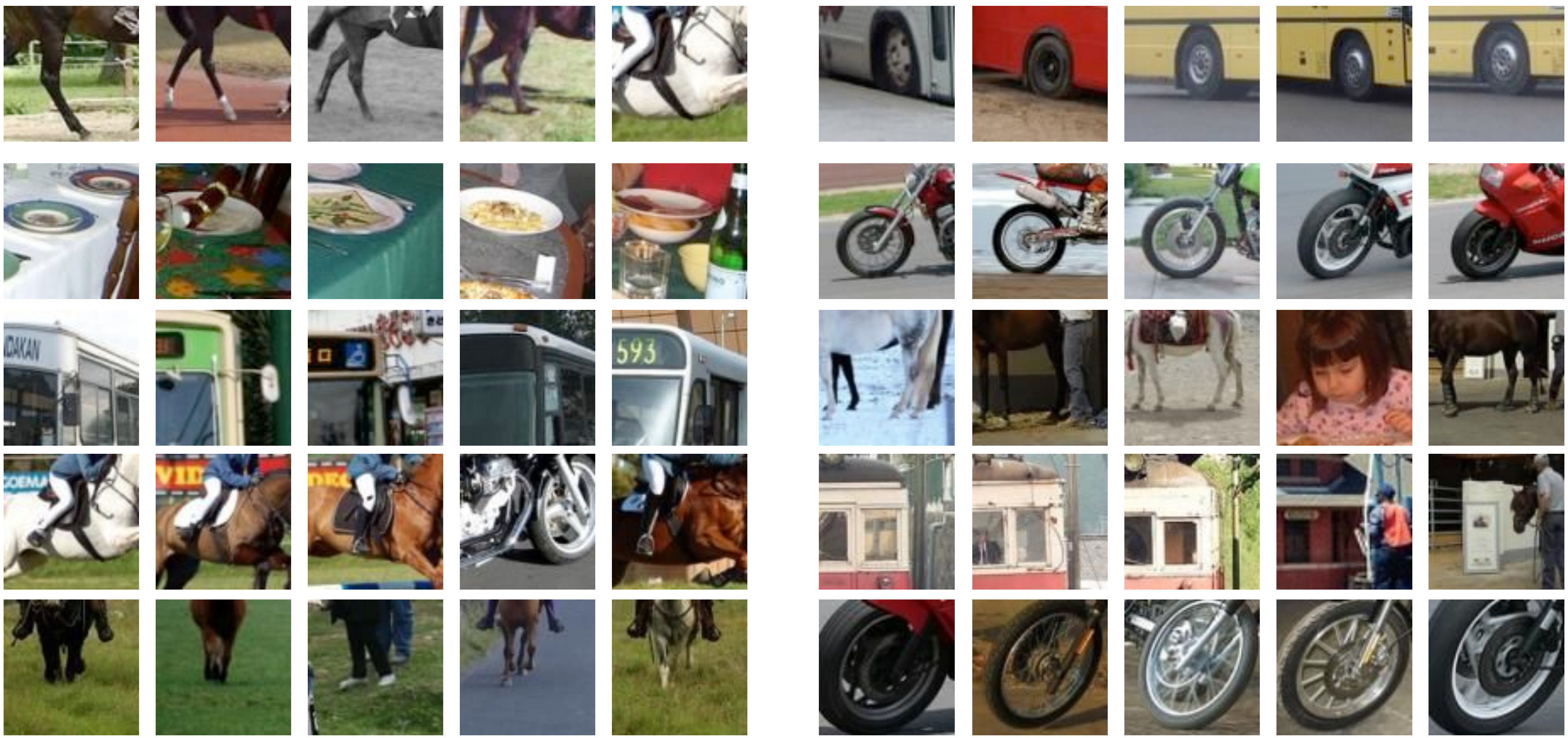
More Results

Image in terms of D+ Patches

Ranking Patches

- Purity: Homogeneity of the clusters. Approximated by the mean SVM score for top few members
- Discriminativeness: How rare are the patches in the "natural world". Approximated by term frequency in "discovery dataset" with respect to both combined.

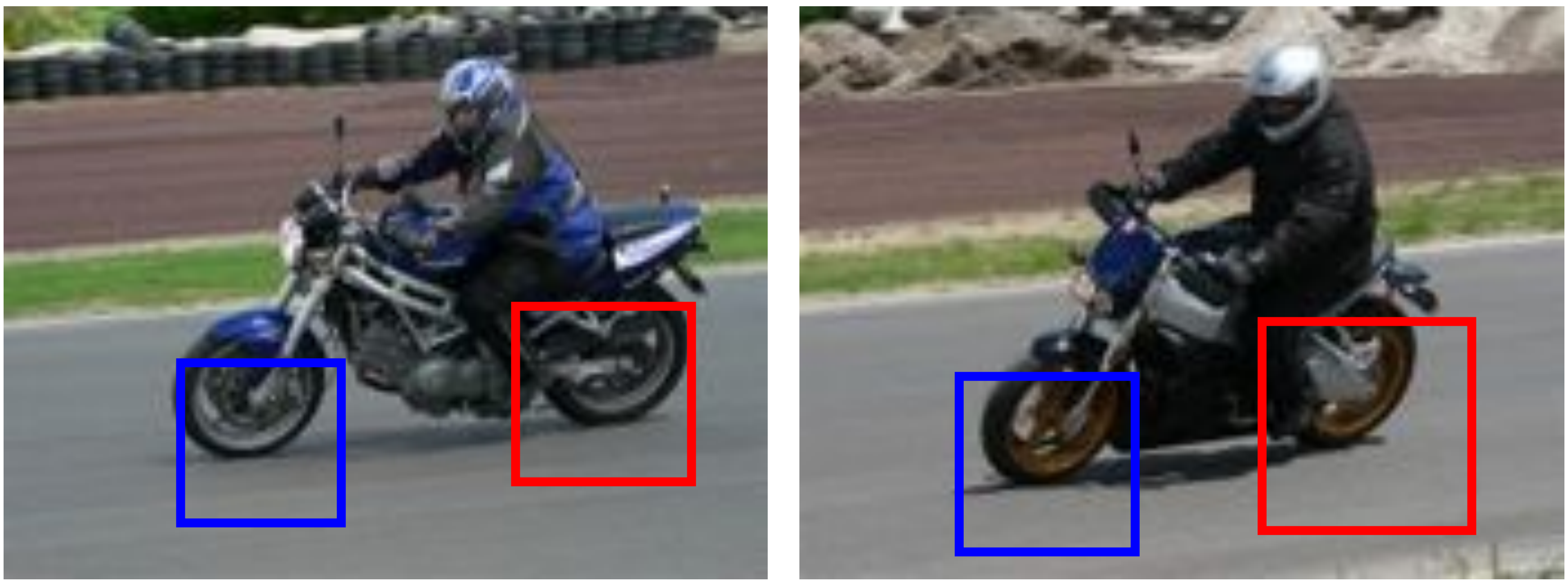
Top Ranked Patches



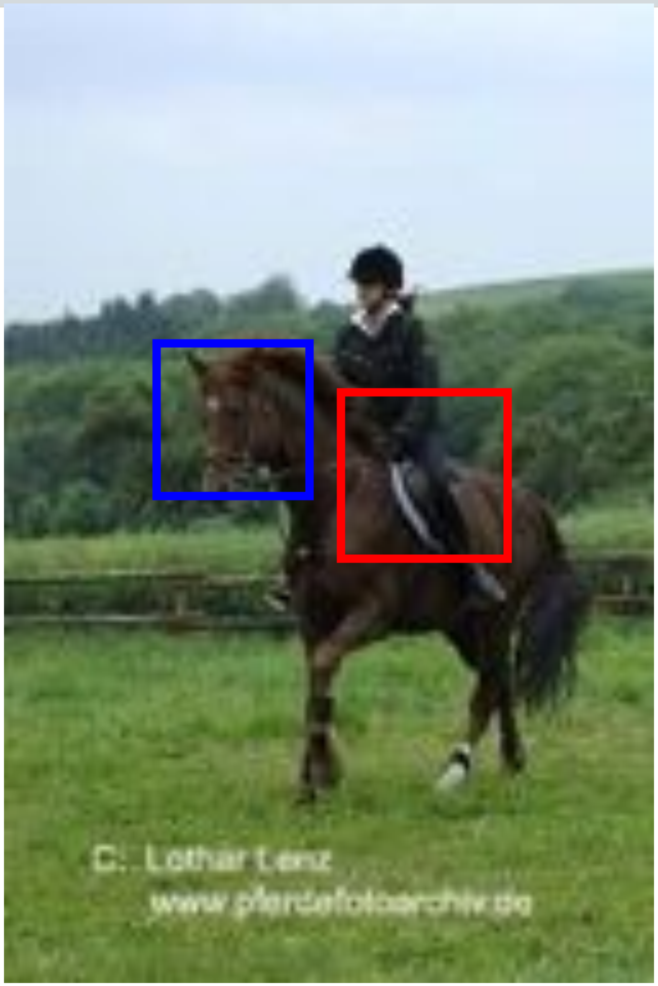
Doublets : Spatially Consistent Pairs

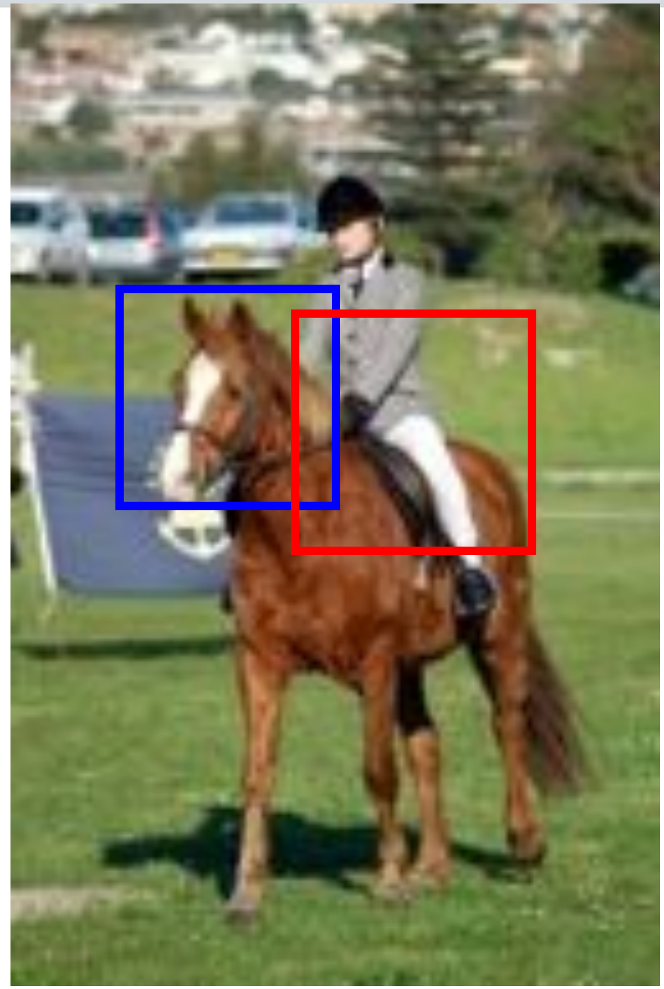
Doublets : Refinement

Discovered Doublets



Discovered Doublets

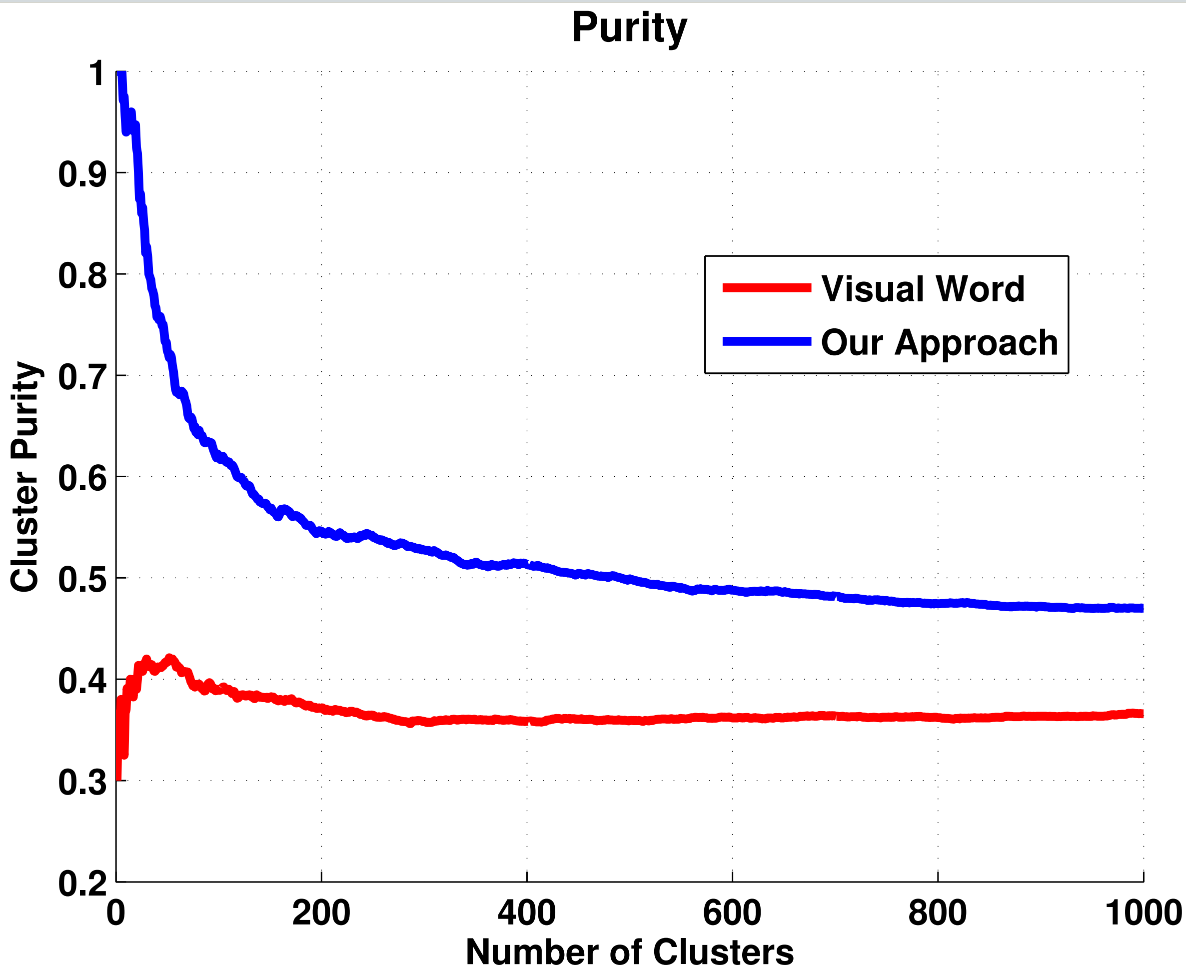




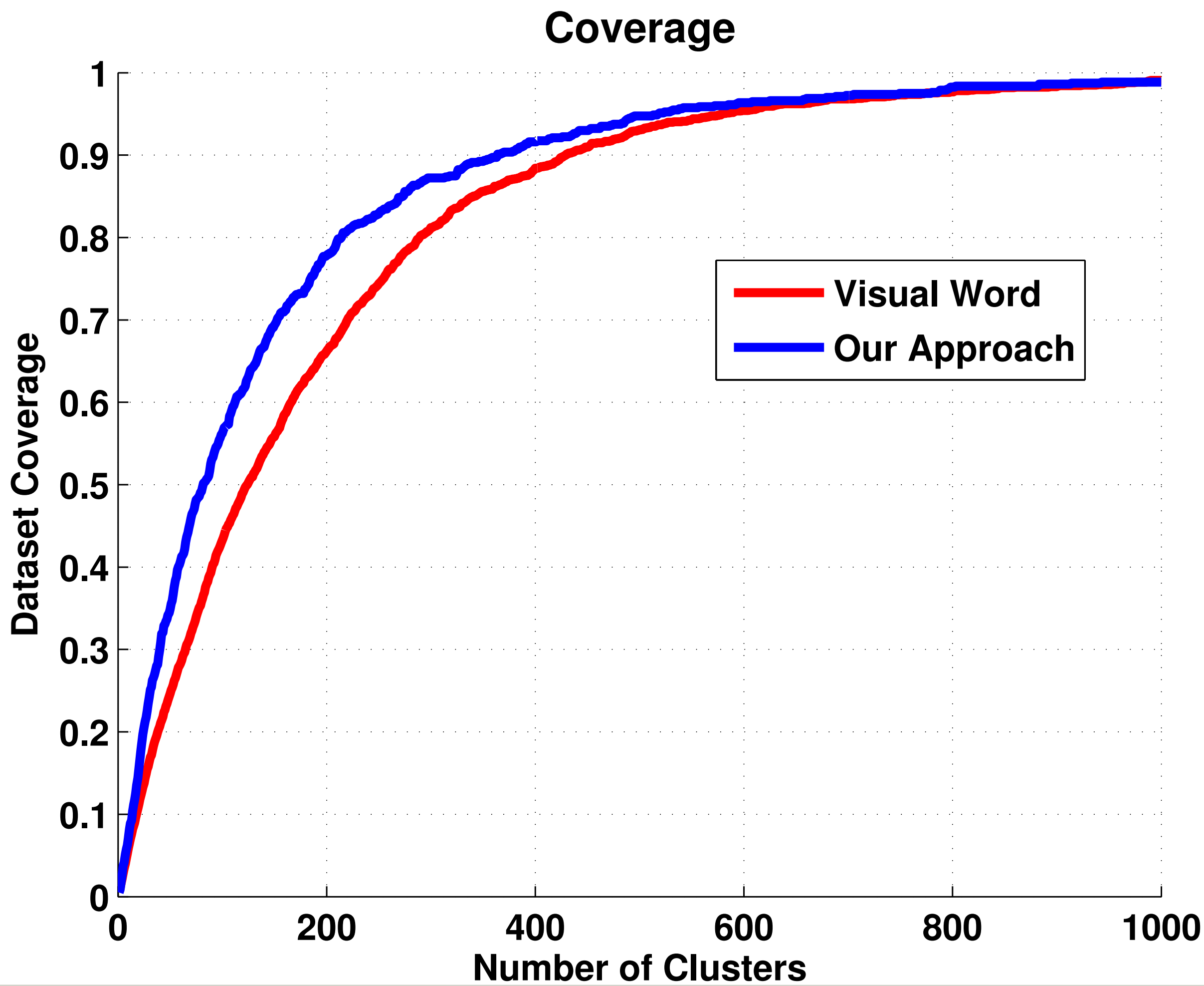
Evaluation

- Comparison with Visual Words
- Dictionary of 1000 visual words to compare against 1000 Discriminative clusters.

Evaluation : Purity



Evaluation : Coverage



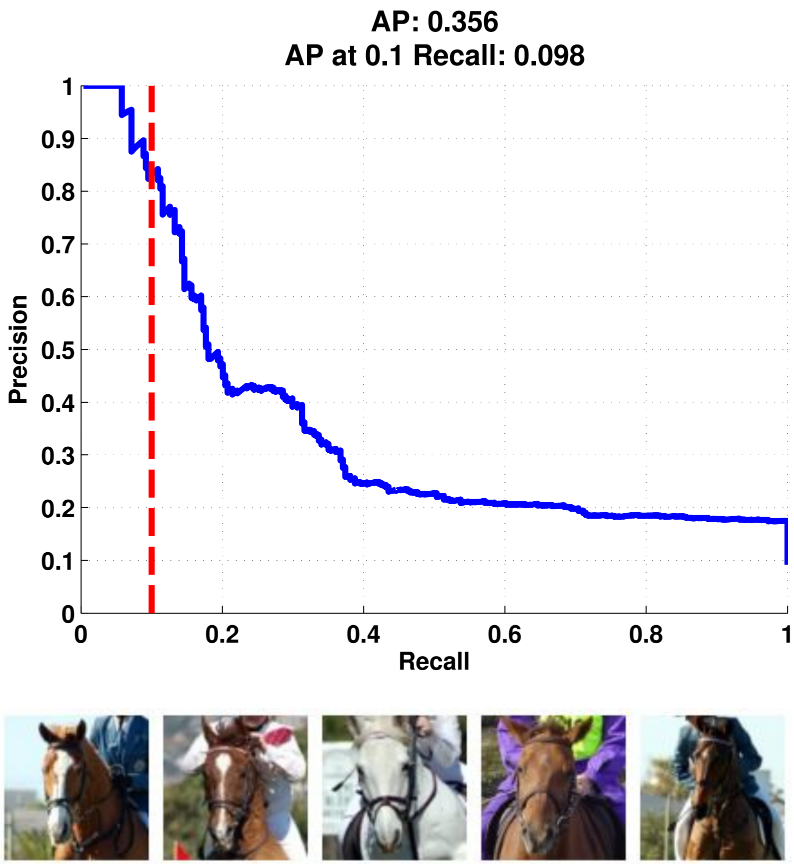
Supervised Image Classification

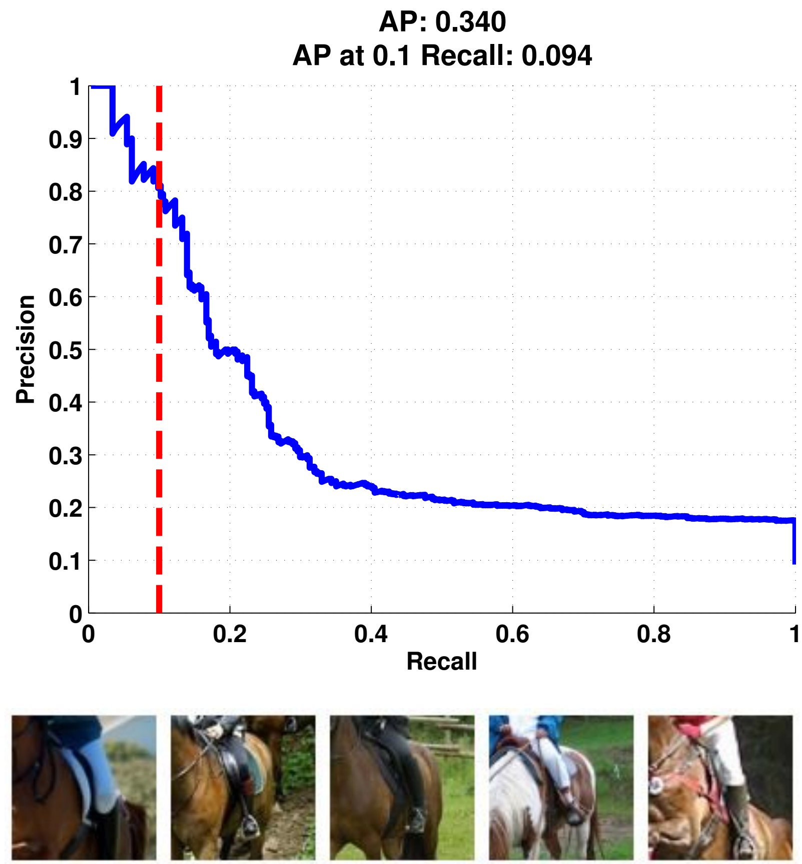
	Bus	Horse	Train	Sofa	Dining Table	Motor Bike	Average
Vis-Word	0.45	0.70	0.60	0.59	0.41	0.51	0.54
D-Pats	0.60	0.82	0.61	0.67	0.55	0.67	0.65
D-Pats + Doublets	0.62	0.82	0.61	0.67	0.57	0.68	0.66

Going Further : More Supervision

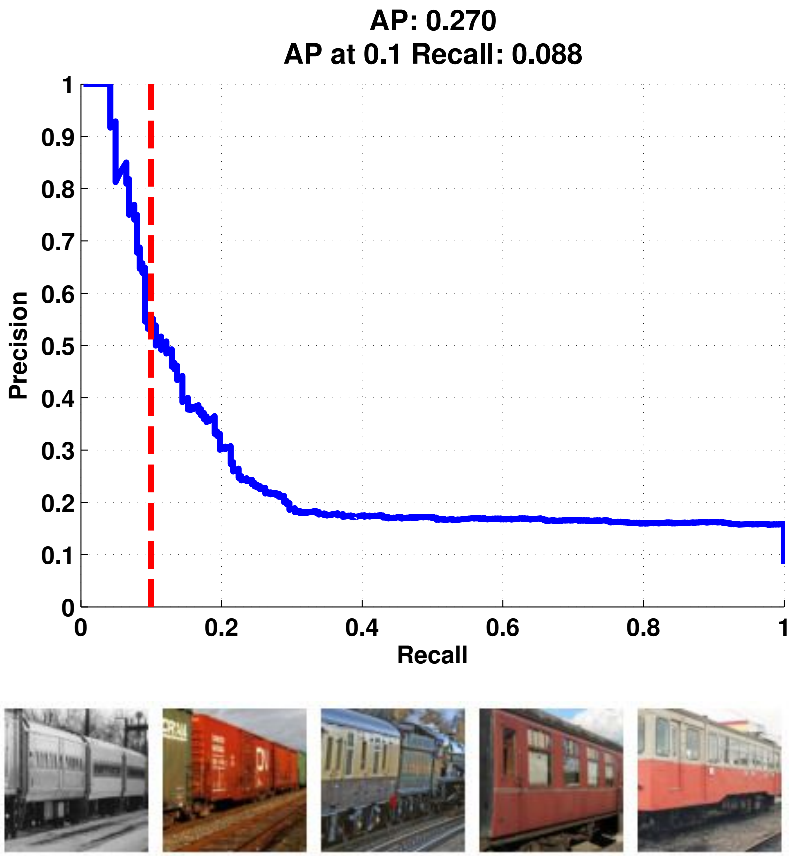
- Discovering using category labels.
- Per-category Clustering.

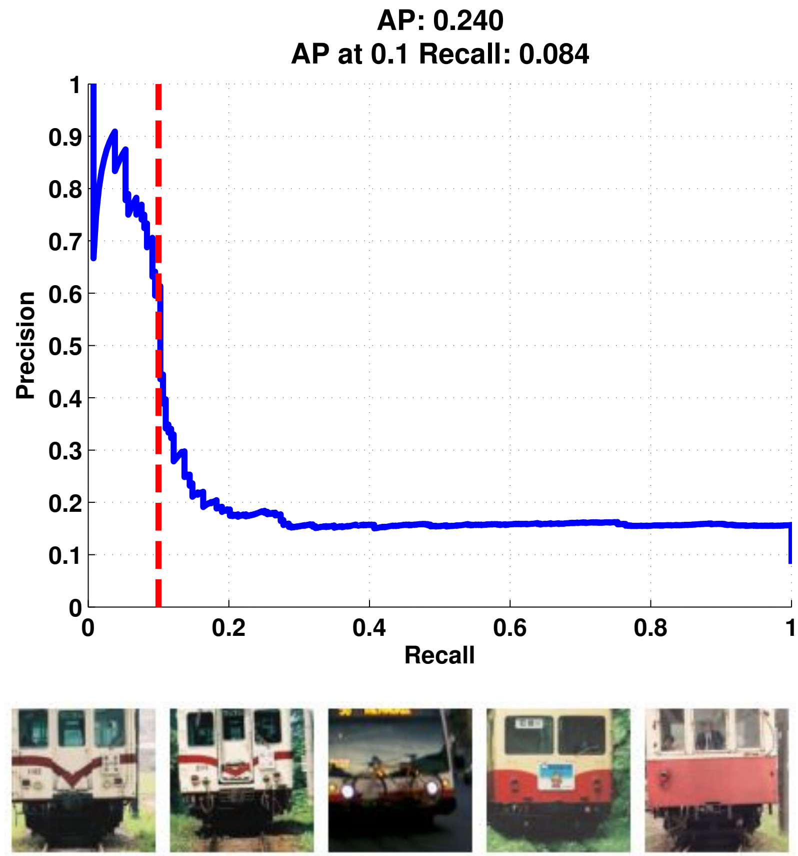
Using Labels





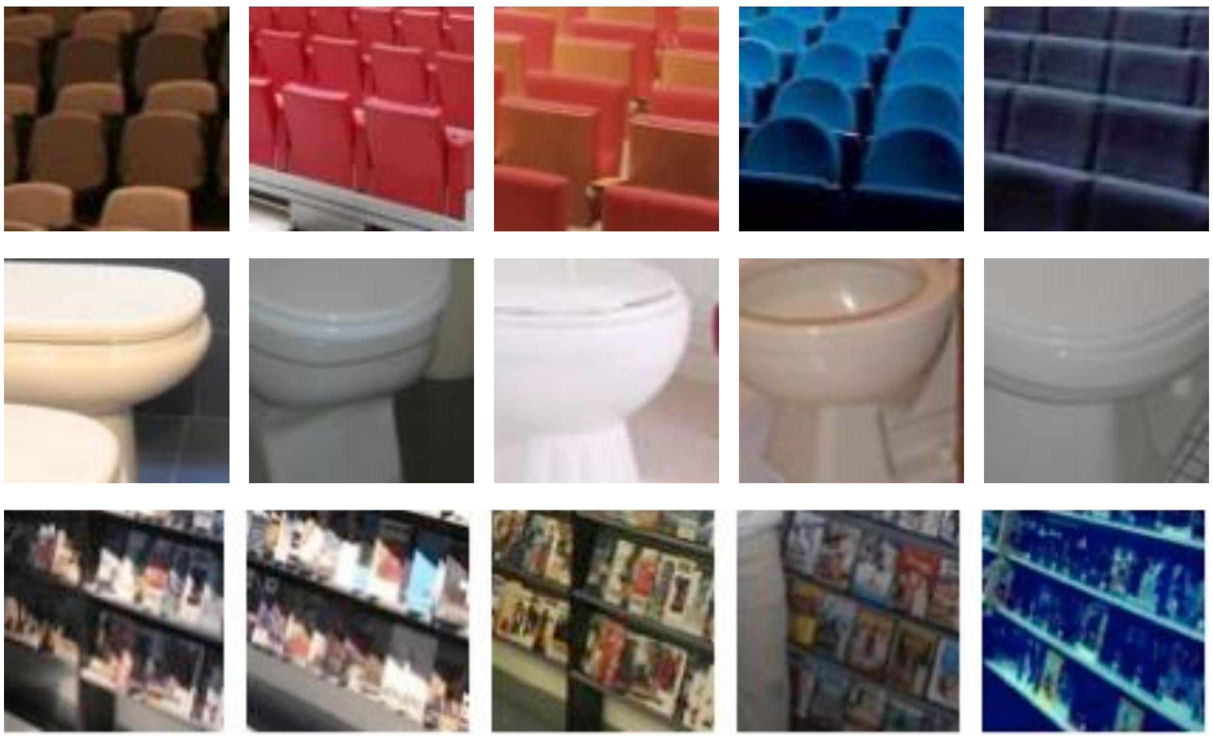
Using Labels





Per-Category Clustering

- Discovery Dataset: Images belonging to a single category



Top Patches Per-Scene

Bookstore

Cloister

Buffet

Bowling

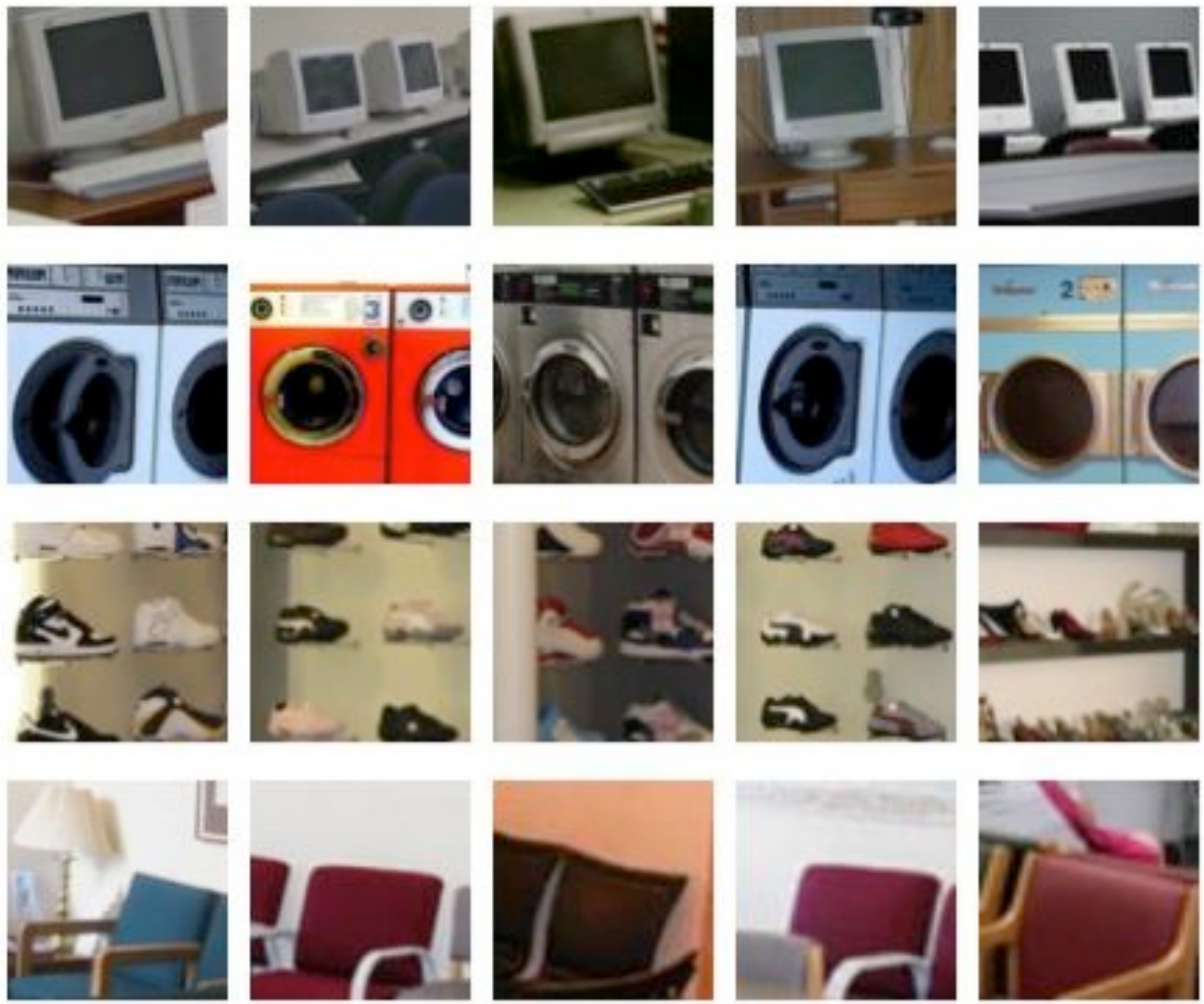
Top Patches Per-Scene

Computer Room

Laundromat

Shoe Shop

Waiting Room

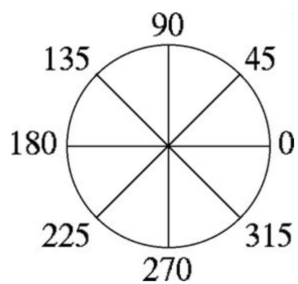


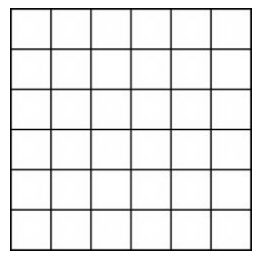
Thank You

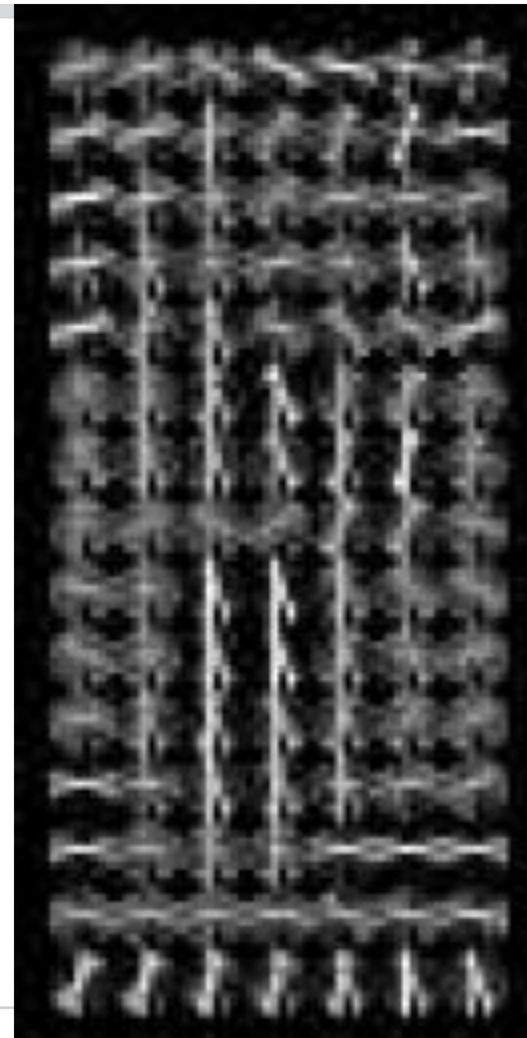
Fun Fact: Only ~300,000 CPU Hours consumed

- Histogram of gradient orientations

  -Orientation -Position







- Weighted by magnitude

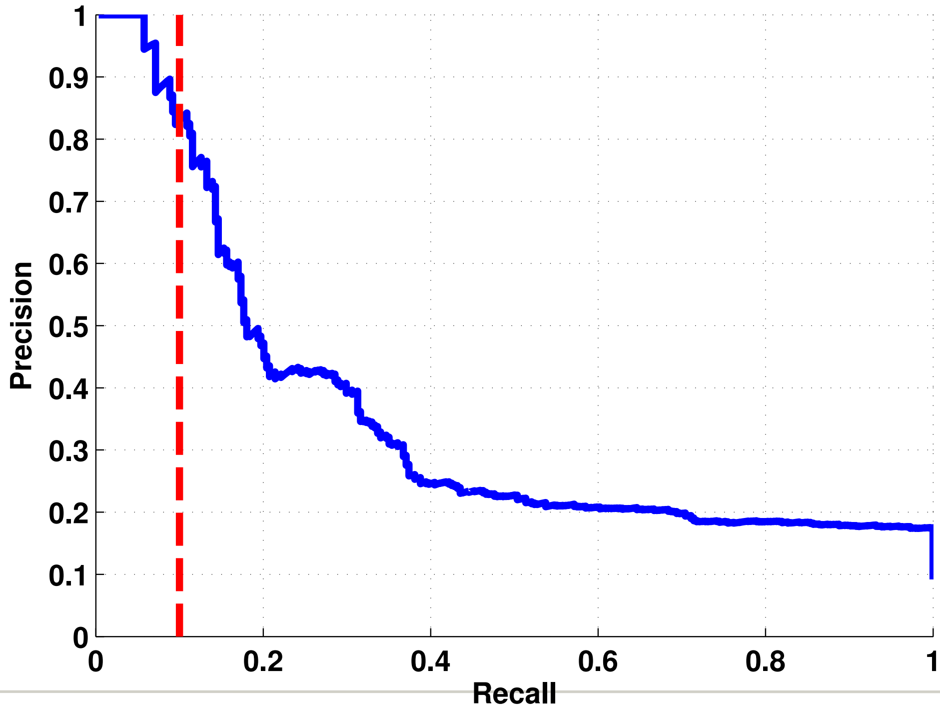
*Borrowed From Alyosha's Slides

Average Precision

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

$$\text{AveP} = \int_0^1 p(r)dr.$$



*Formulas from Wikipedia

Spatial Pyramid