Stealing Objects With Computer Vision

Learning Based Methods in Vision
Analysis Project #4: Mar 4, 2009
Presented by: Brian C. Becker
Carnegie Mellon University

Motivation

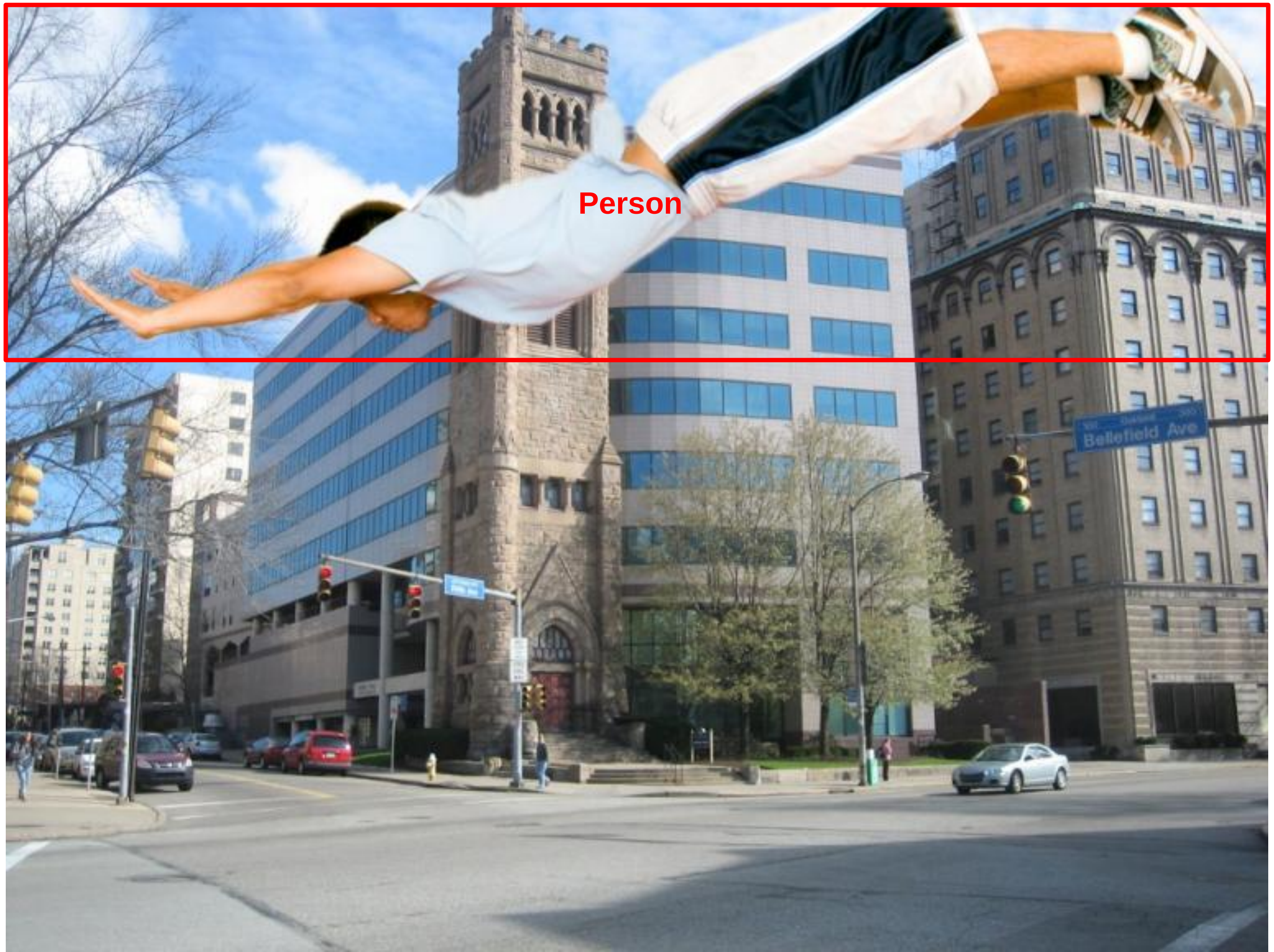
- Goal: Detect objects in the photo you just took

Motivation

- Scanning window

Motivation

Motivation



Motivation

Motivation

Motivation

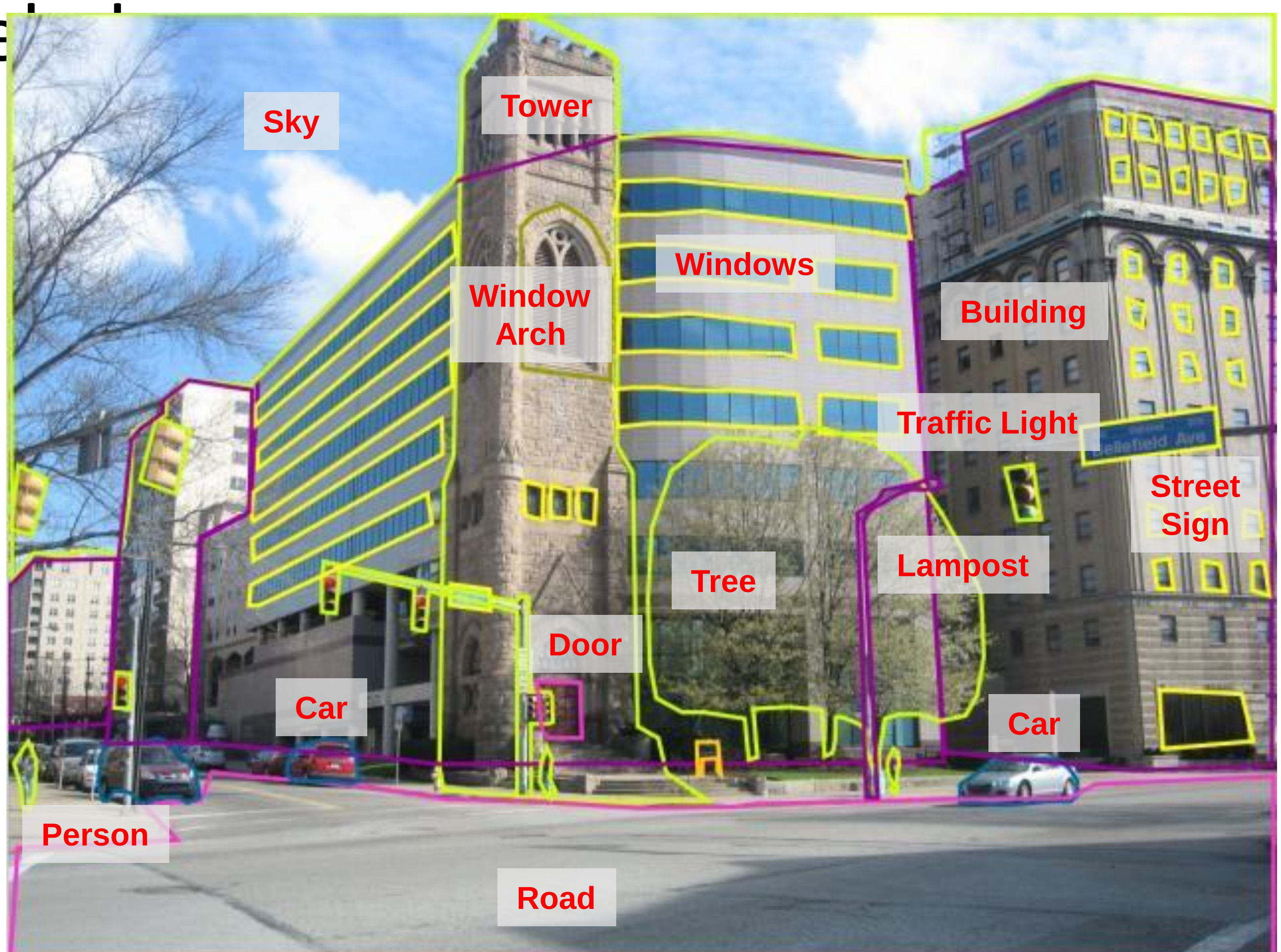
Motivation

Motivation

- What else can we try for object recognition?

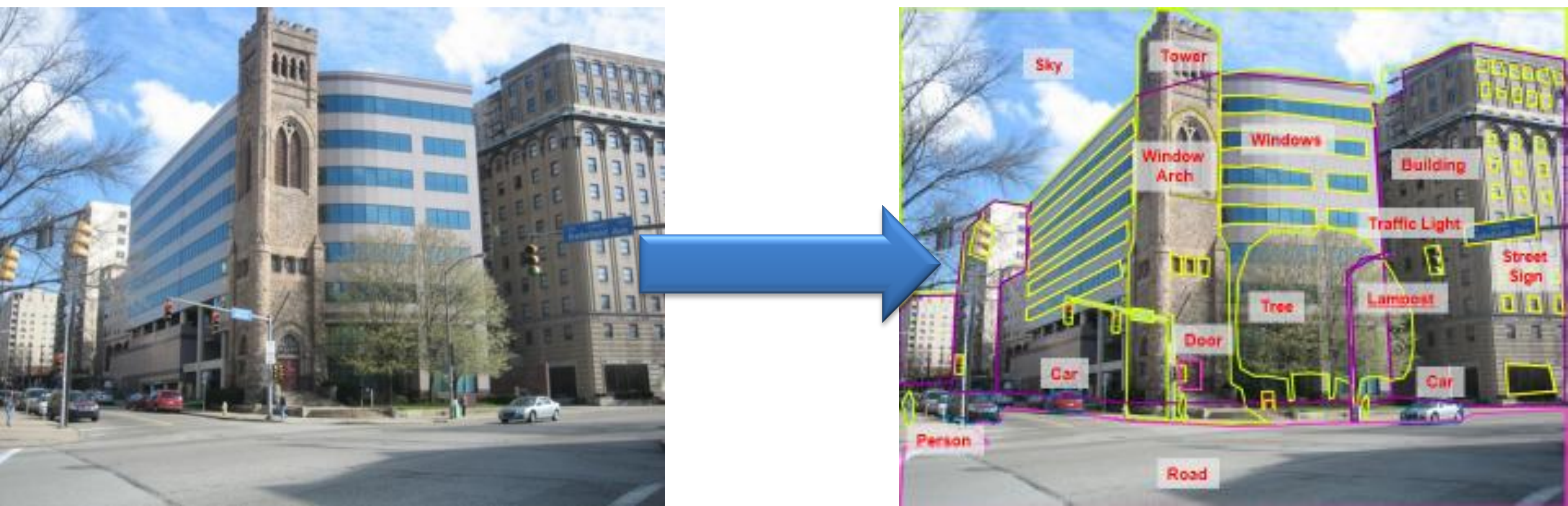
Object Detection

- Go to internet and behold! exact picture labe



Object Detection

- Ideally, object detection is giant lookup
  - Labeled plenoptic function
  - Label everything in the world from all viewpoints
- Labelme: Online annotation tool



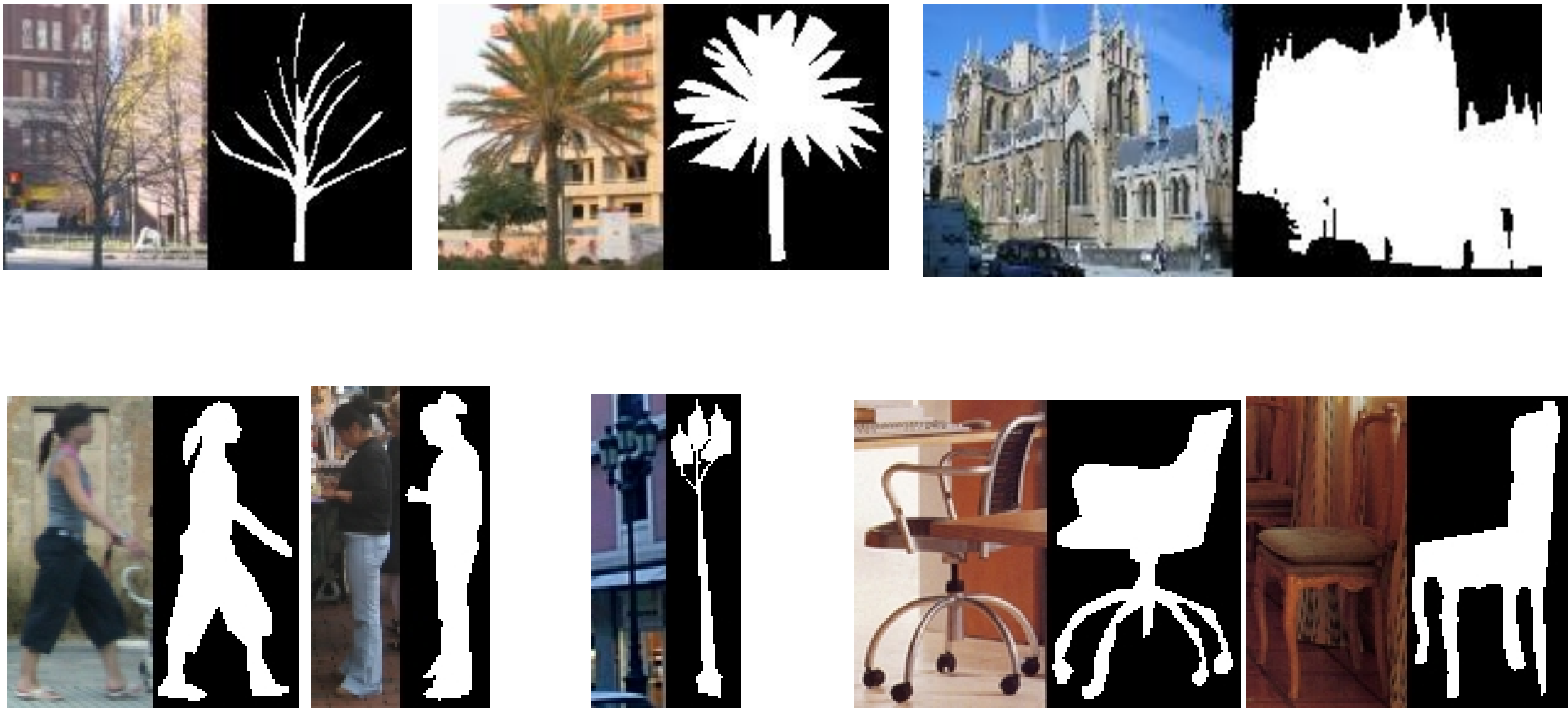
Tool went online July 1st, 2005

290,000 object annotations

Labelme.csail.mit.edu

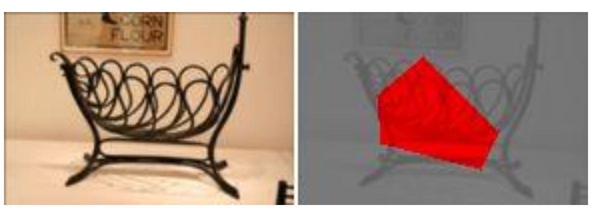
B. Russell, A. Torralba, K. Murphy, W.T. Freeman. IJCV 2008

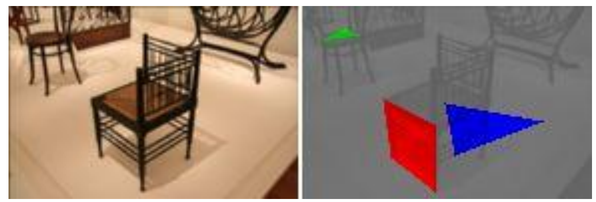
Labelme Polygon Quality



Labelme Polygon Diversity

Labelme Testing





Most common labels:

test

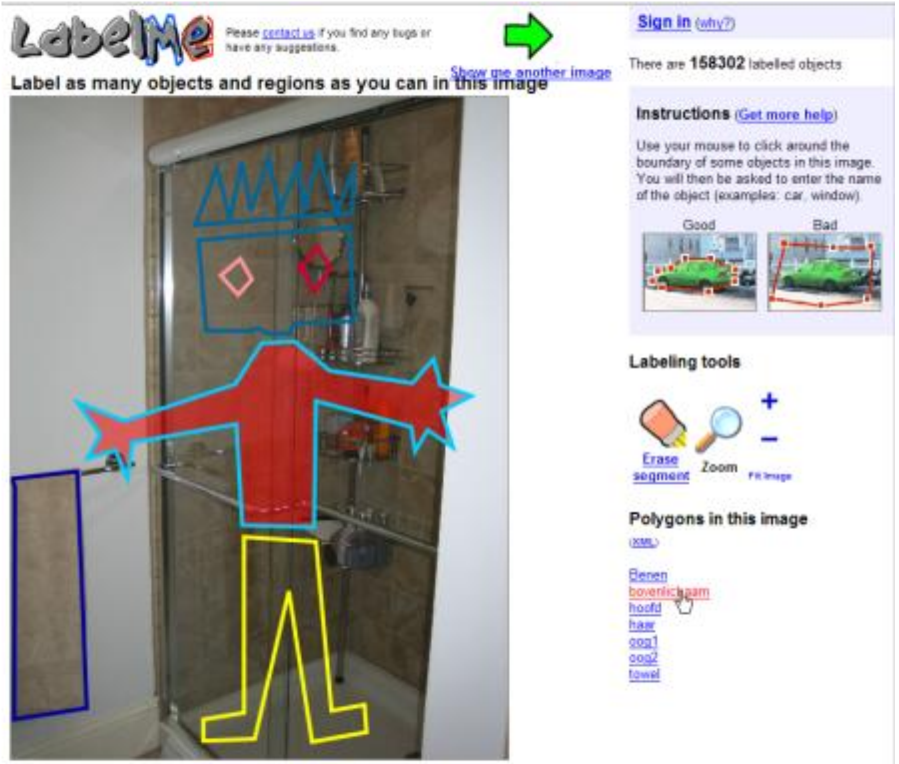
adksdsa

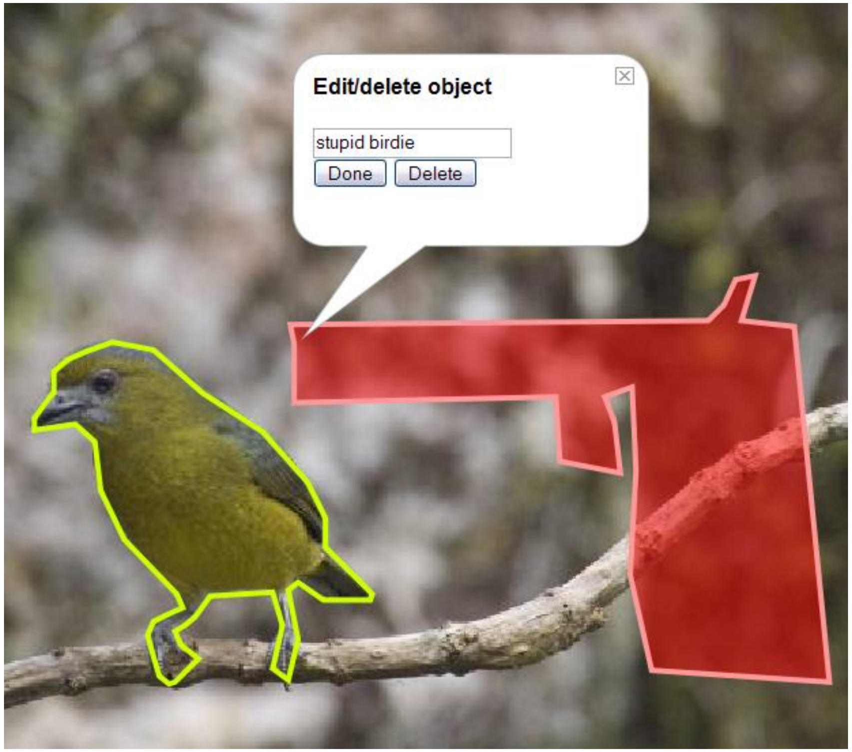
woiieiie

…

Labelme Hooligans

Do not try this at home





Labelme Database

- 30 GB dataset of
  - 176,000 photos total
  - 52,000 photos with annotations

Labelme Matlab Toolbox

`LMquery (database, 'object.name', 'car,building,road,tree')`

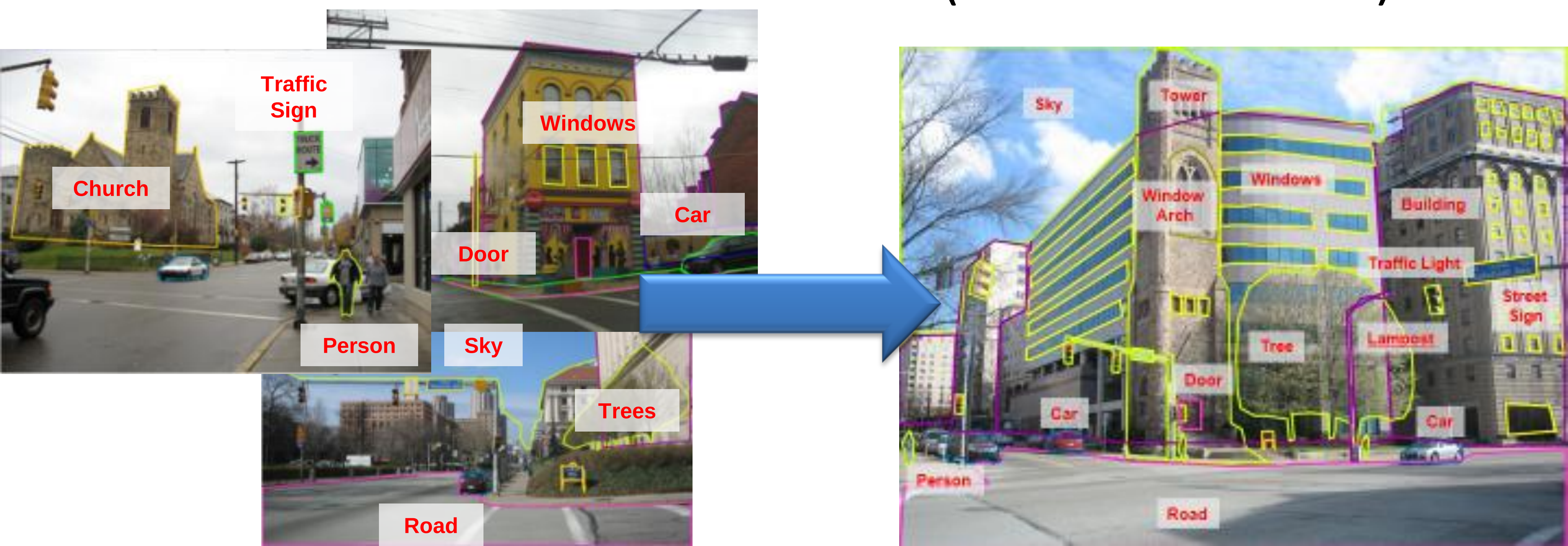
- Query objects
- Extract polygons
- Annotation stats
- Label merging
- Wordnet reasoning
- Manipulate images
- Scene descriptors

Wordnet Object & Parts

Labelme Average Objects

Object Detection

- Unfortunately, Labelme is not God
- Next best thing
  - Find similar scenes containing similar objects
  - Steal information from them (i.e. label transfer)



Papers

- SIFT Flow Paper
  - C. Liu, J. Yuen, A. Torralba, J. Sivic, W.T. Freeman. "SIFT Flow: Dense Correspondence across Different Scenes." ECCV 2008.
- Context Paper
  - B. C. Russell, A. Torralba, C. Liu, R. Fergus, W.T. Freeman. "Object Recognition by Scene Alignment." NIPS 2007.

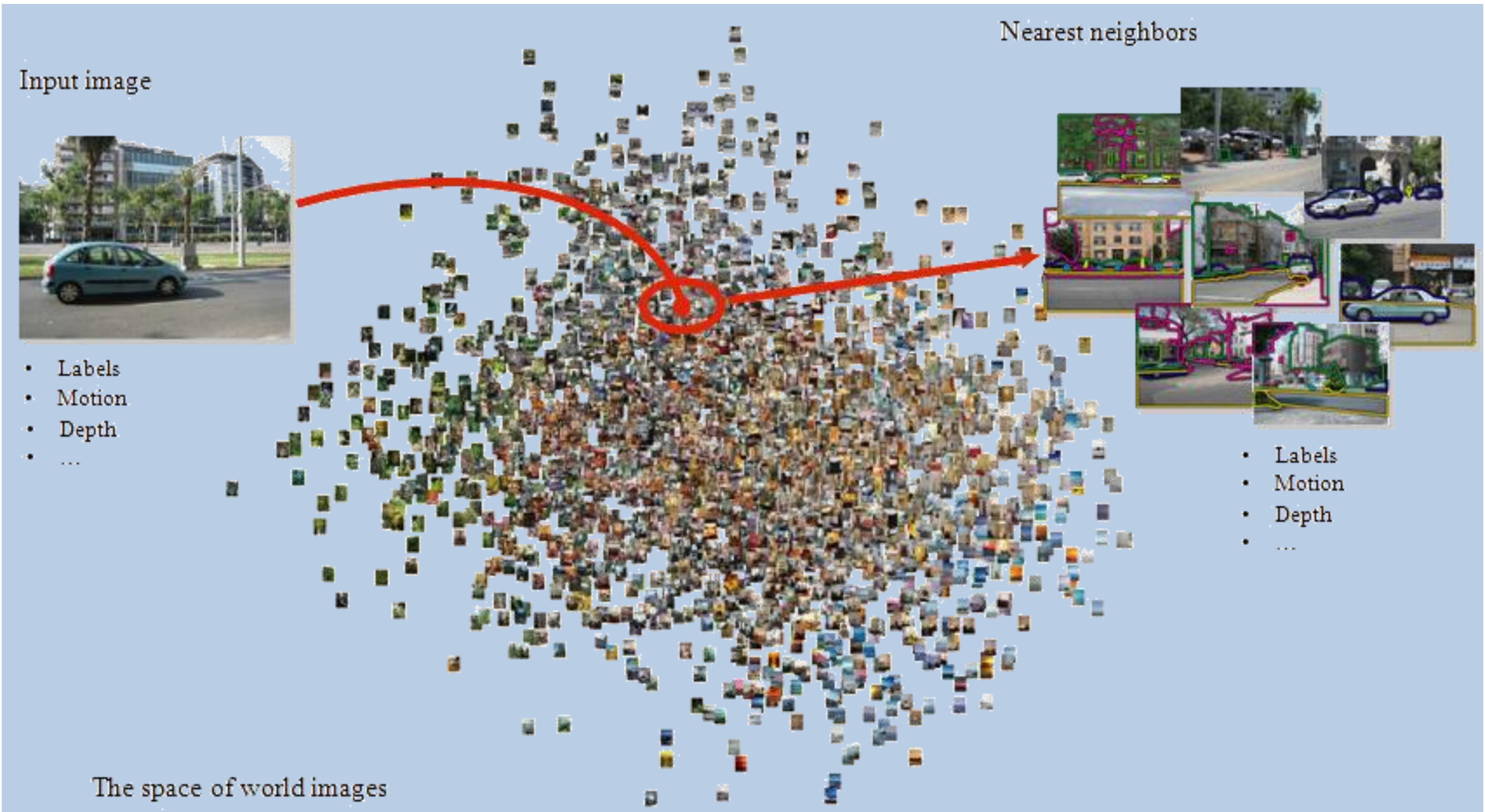
SIFT Flow

- SIFT Flow Goal: Align objects in similar scenes
  - Problem: Current alignment algorithms aren't robust
  - Solution: SIFT is magic and works, find the flow of image patches to a similar image

- If your dataset isn't infinite, find a close match and rearrange (wiggle) it so it is aligned

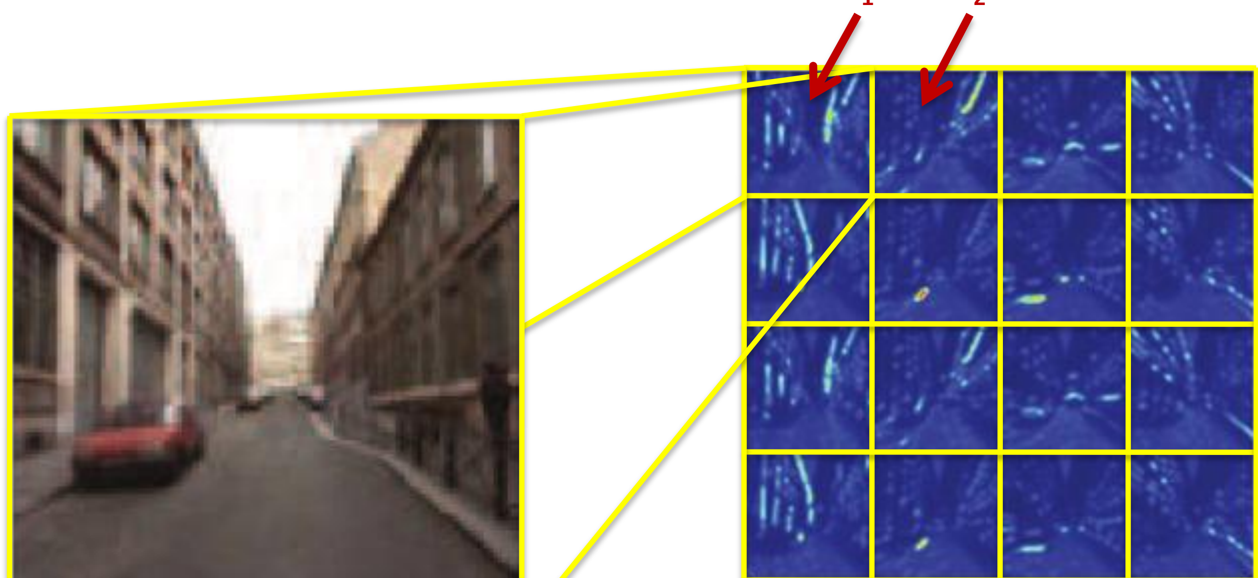
- SIFT Flow "allows matching of objects located at different parts of the scene"

SIFT Flow


Nearest neighbors
Input image
• Labels
• Motion
• Depth
• …
• Labels
• Motion
• Depth
• …
The space of world images

Matching SIFT Features

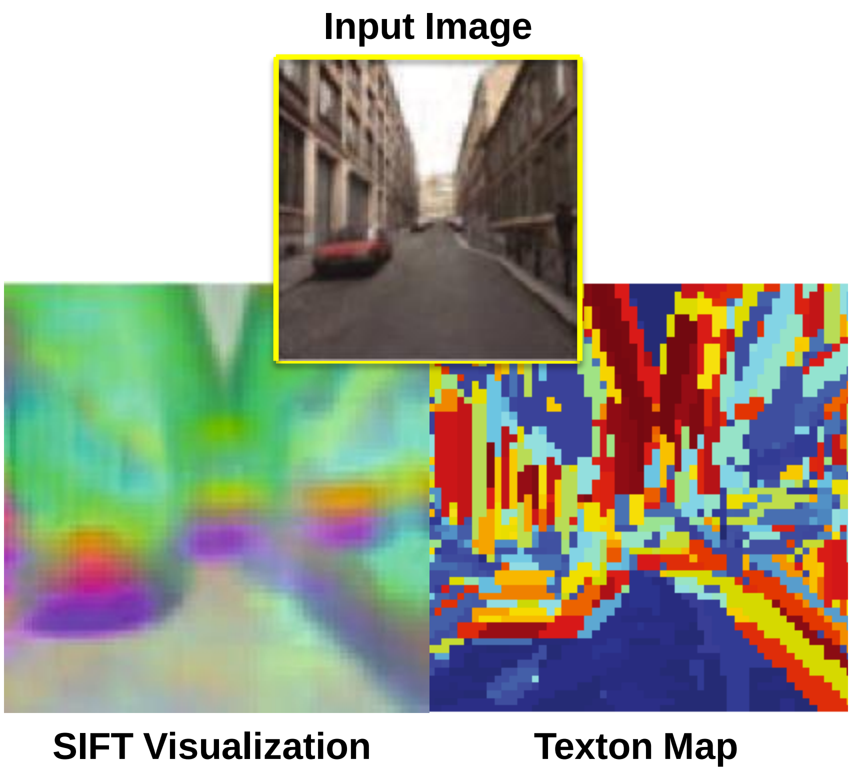
- Decompose image into scene descriptors
- SIFT features (D. Lowe, 1999)
  - 128 dimensional vector (u_1, ..., u_{128}) at each pixel



Input Image

First 16 dimensions of SIFT descriptor

Matching SIFT Features

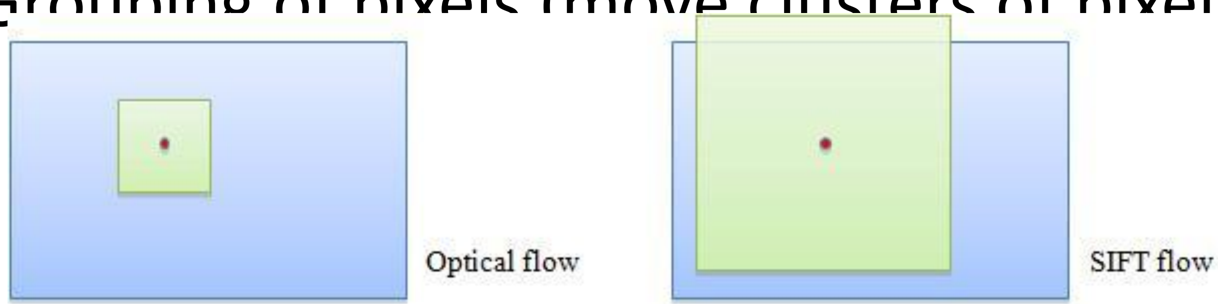


- Use "bag-of-words" to cluster SIFT features into 500 visual words
  - Good ole K-means
- Reduce image to texton map of SIFT features

- Fast/coarse matching on SIFT texton map
- Top 20 fast matches re-ranked with SIFT Flow

SIFT Flow

- Optical flow without spatial limitations
- Assumptions:
  – SIFT descriptors at each pixel are constant with respect to the pixel displacement field
  – One pixel may move as much as the size of the image
  – Grouping of pixels (move clusters of pixels)



SIFT Flow

- Formulate as an optimization problem

$$E(\mathbf{w}) = \sum_{\mathbf{p}} \left\| s_1(\mathbf{p}) - s_2(\mathbf{p}+\mathbf{w}) \right\|_1 + \frac{1}{\sigma^2} \sum_{\mathbf{p}} \left(u^2(\mathbf{p}) + v^2(\mathbf{p}) \right) +$$
$$\sum_{(\mathbf{p},\mathbf{q}) \in \varepsilon} \min\left(\alpha |u(\mathbf{p}) - u(\mathbf{q})|, d \right) + \min\left(\alpha |v(\mathbf{p}) - v(\mathbf{q})|, d \right),$$

  - **w**(**p**)=(u(**p**),v(**p**)) is the displacement vector at pixel location **p**=(x,y)
  - S_i(p) is the SIFT descriptor extracted at location **p** in image *I*
  - *E* is the spatial neighborhood of a pixel

SIFT Flow

- Formulate as an optimization problem

How close the matched SIFT descriptors are

SIFT feature at (x,y) in image 1

Matched SIFT feature at (x+u,y+v) in image 2

Add a cost for large displacements

$$E(\mathbf{w}) = \sum_{\mathbf{p}} \left\| s_1(\mathbf{p}) - s_2(\mathbf{p}+\mathbf{w}) \right\|_1 + \frac{1}{\sigma^2} \sum_{\mathbf{p}} \left(u^2(\mathbf{p}) + v^2(\mathbf{p}) \right) + \sum_{(\mathbf{p},\mathbf{q}) \in \varepsilon} \min\left(\alpha |u(\mathbf{p}) - u(\mathbf{q})|, d \right) + \min\left(\alpha |v(\mathbf{p}) - v(\mathbf{q})|, d \right),$$

Model discontinuities

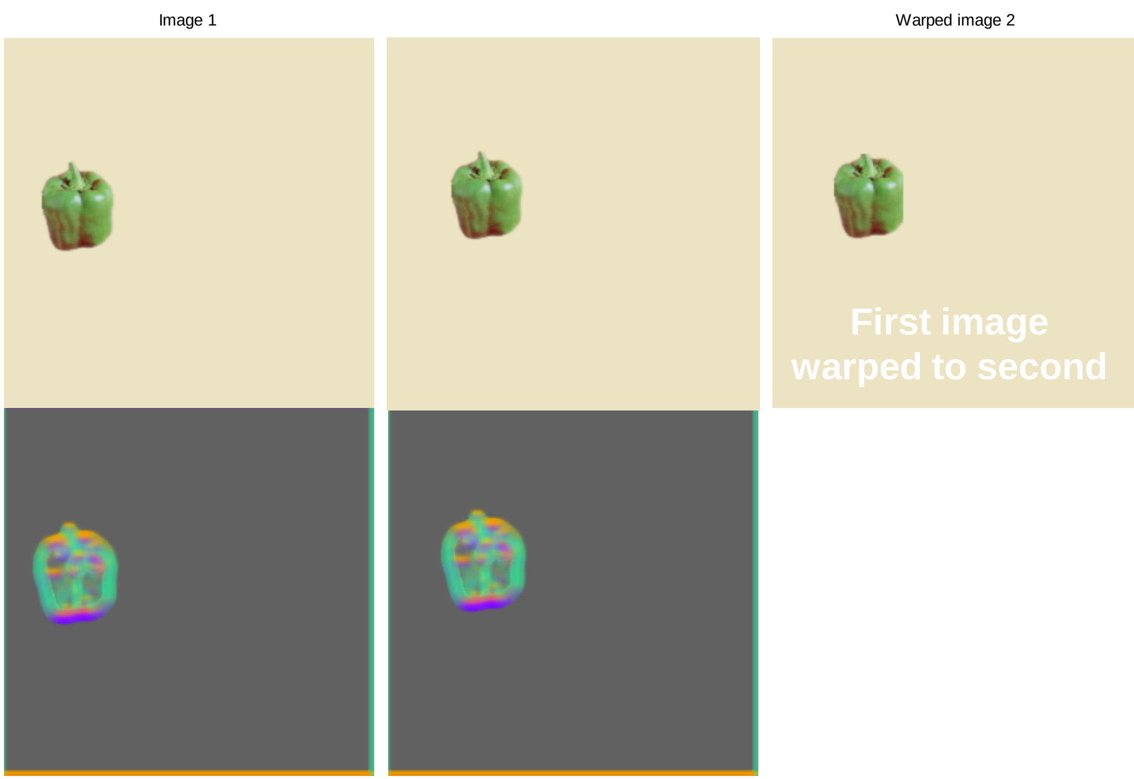
  - u and v are decoupled to reduce complexity from $O(L^3)$ to $O(L^2)$. L is the size of the search window.

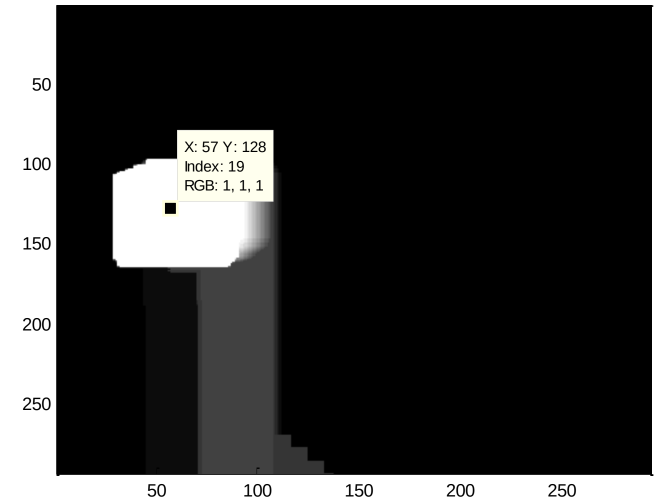
SIFT Flow Example

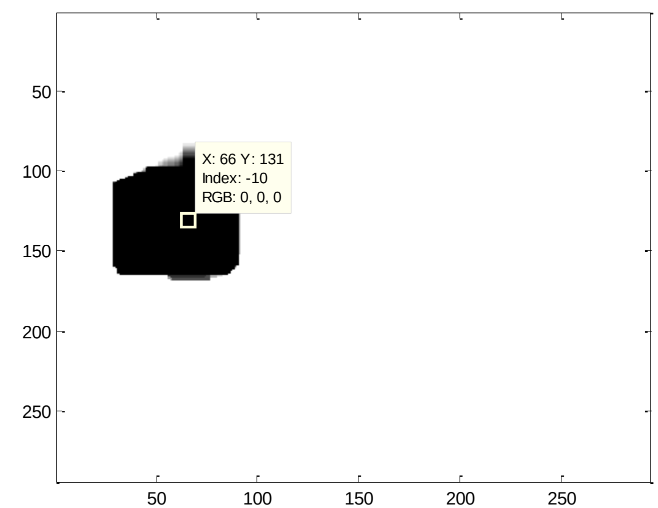
- SIFT Flow “allows matching of objects located at different parts of the scene”
- Hypothesis: Pixels from an object in one image will “flow” to the same class of objects in a second image
- Let’s test that with a simple example

SIFT Flow Pepper Example

- Two images of a pepper
  - One pepper is shifted 20 pixels right, 10 pixels up

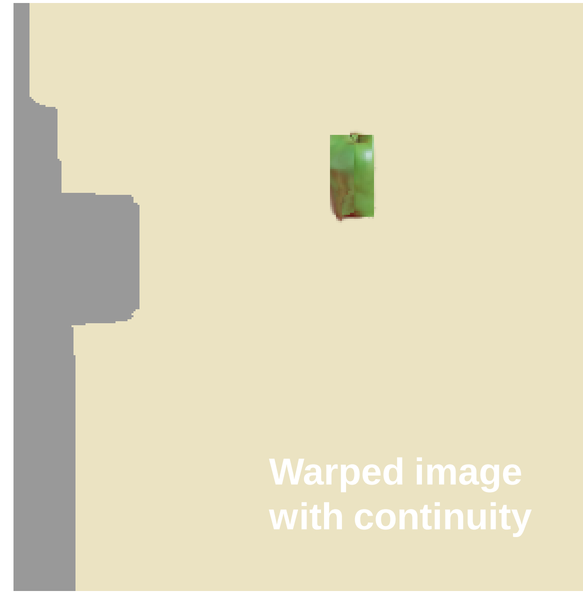




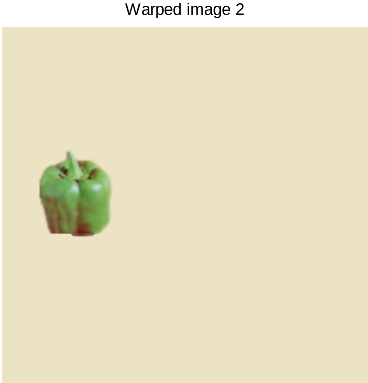


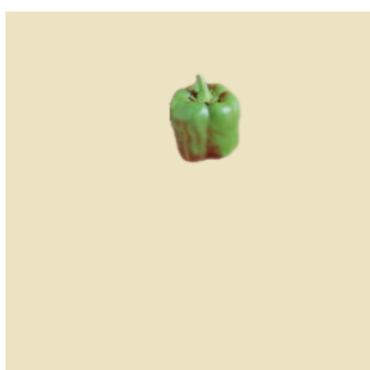
SIFT Flow Pepper Example

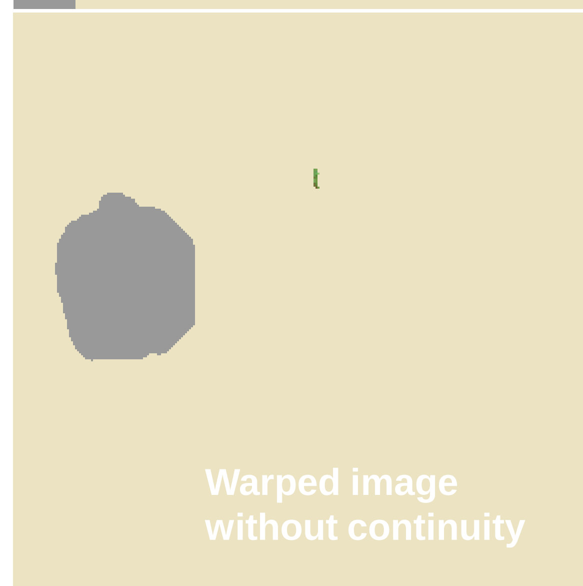
- Two images of a pepper
  - One pepper is shifted 100 pixels right, 50 pixels up
- Test turning off continuity
- Needs lot of tweaking



Warped image 2



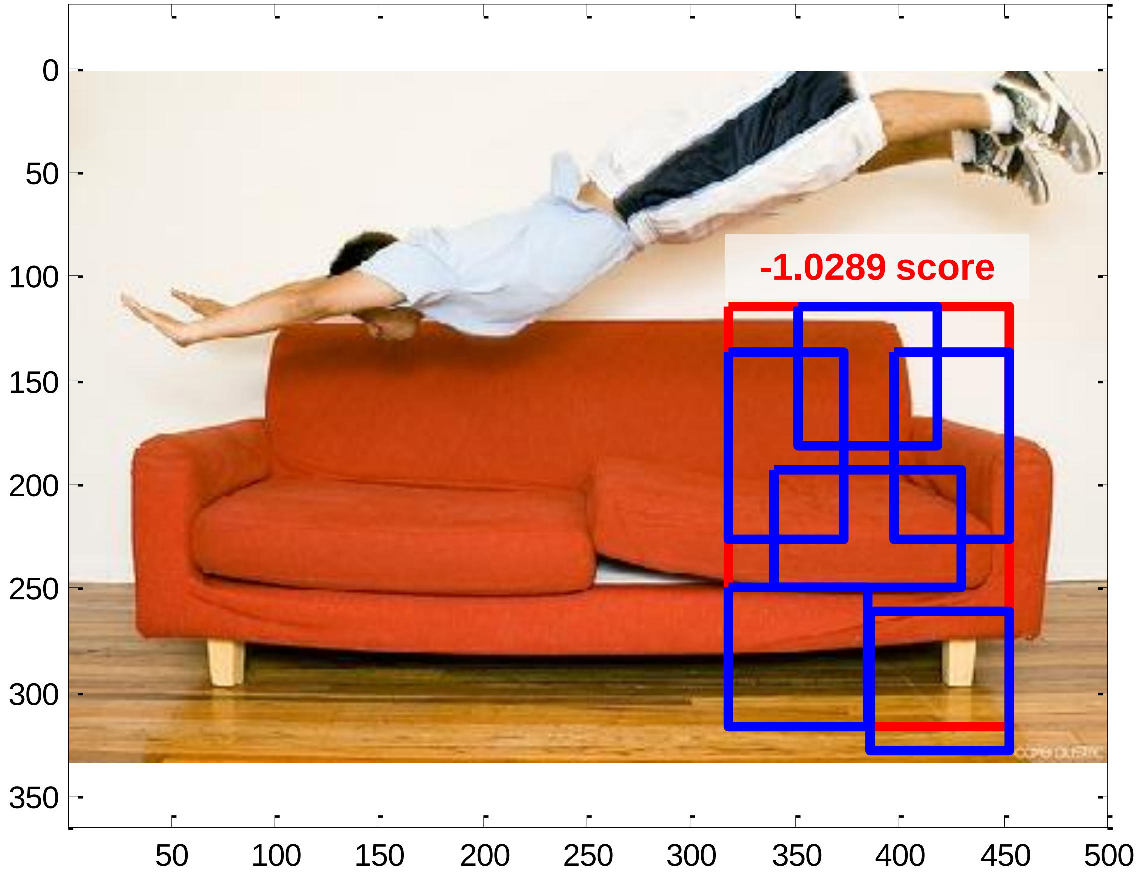




SIFT Flow Hard Example

SIFT Flow Hard Example

- Felzenszwalb parts-based HOG detector says

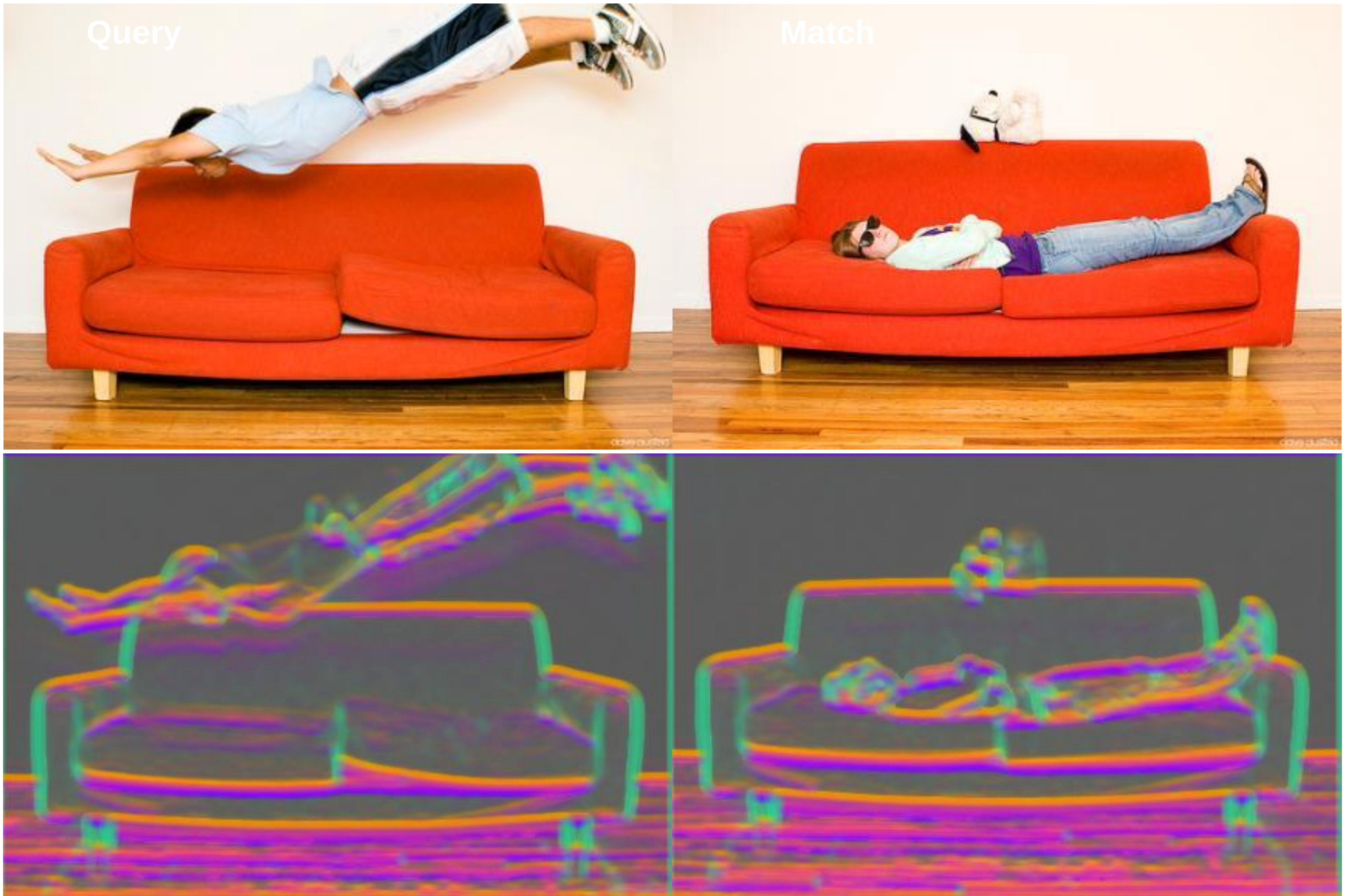


SIFT Flow Hard Example

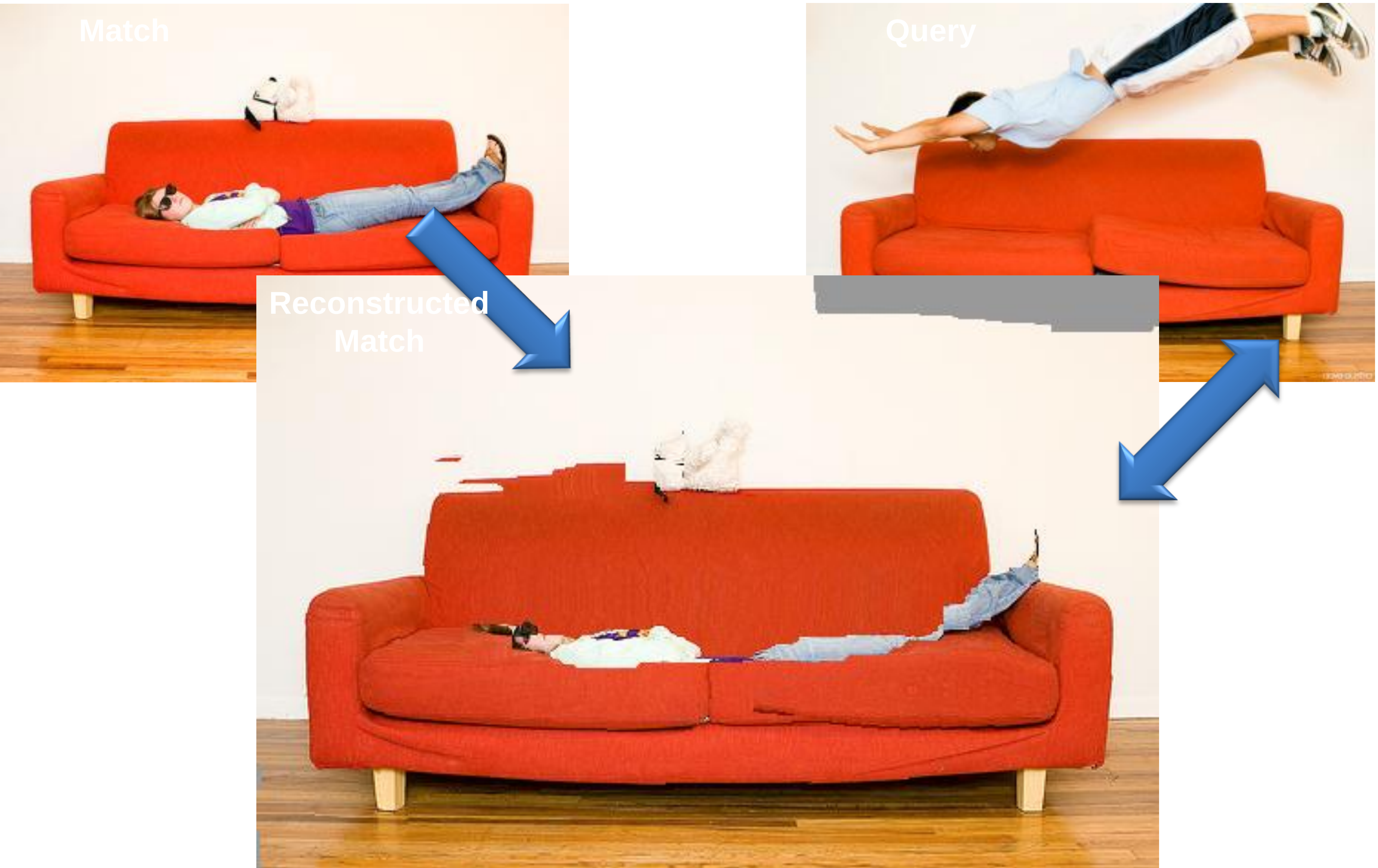
SIFT Flow Hard Example

- Best match, most similar labeled photo

SIFT Flow Hard Example

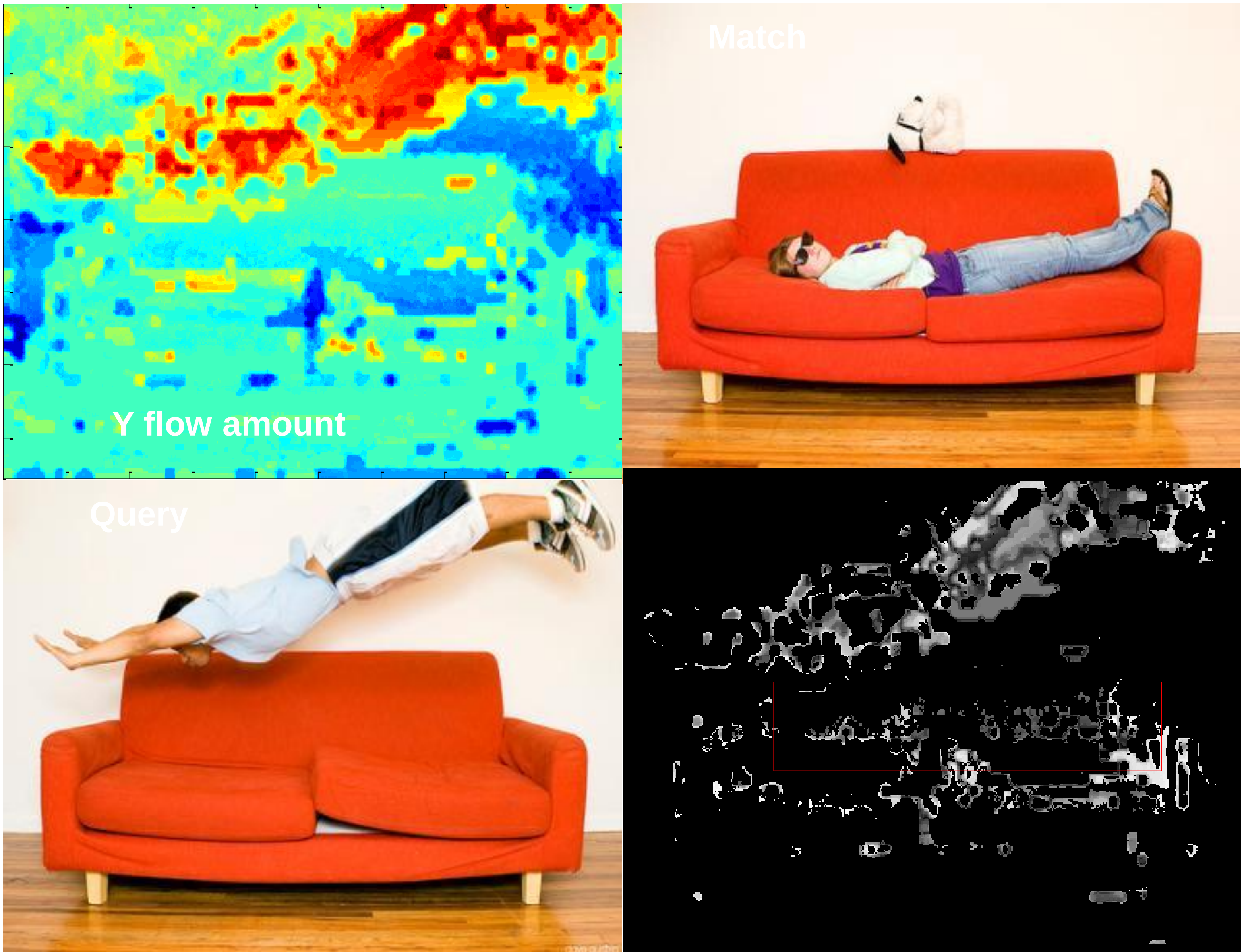


SIFT Flow Hard Example

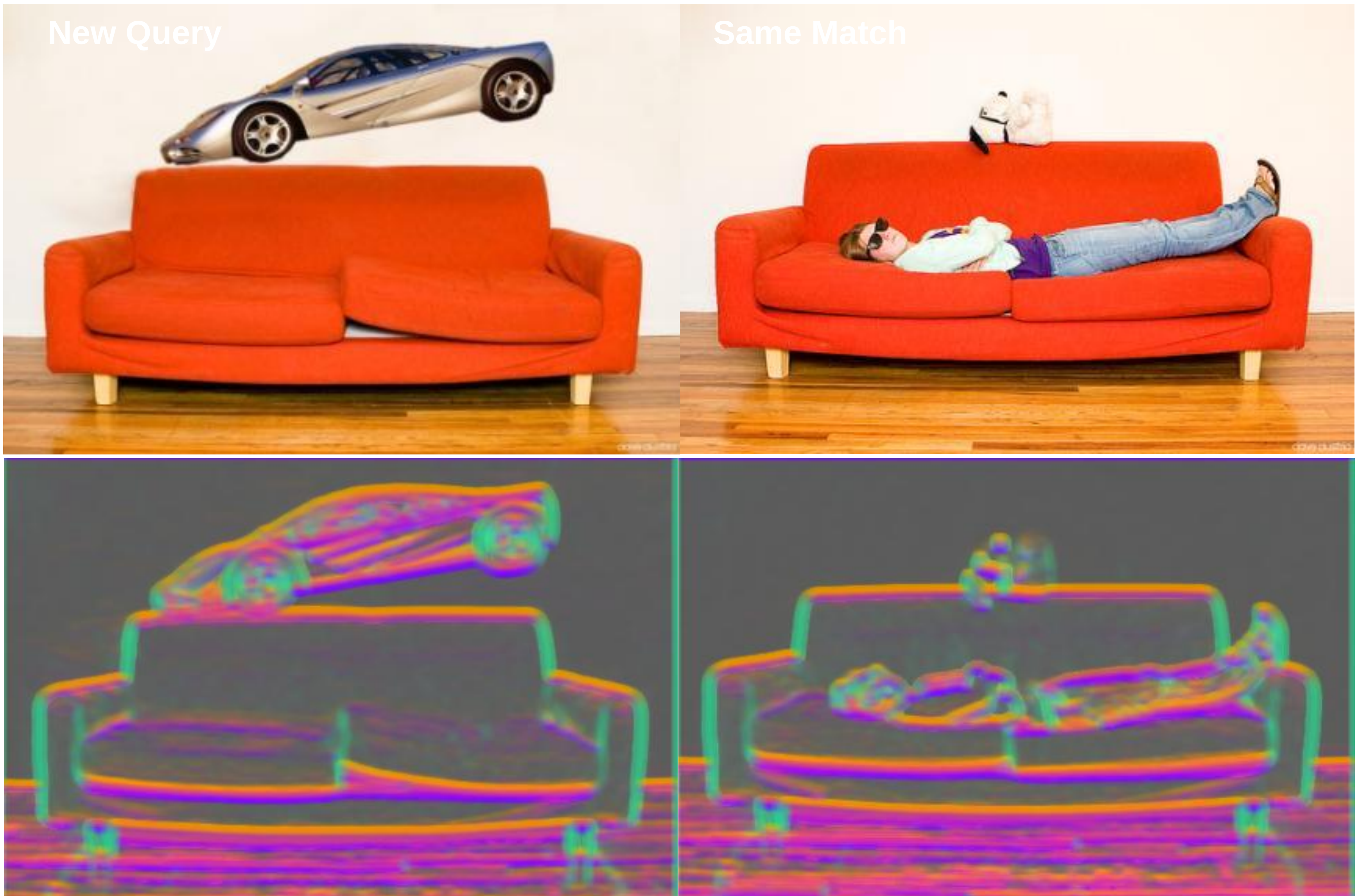


SIFT Flow Hard Example

SIFT Flow Hard Example

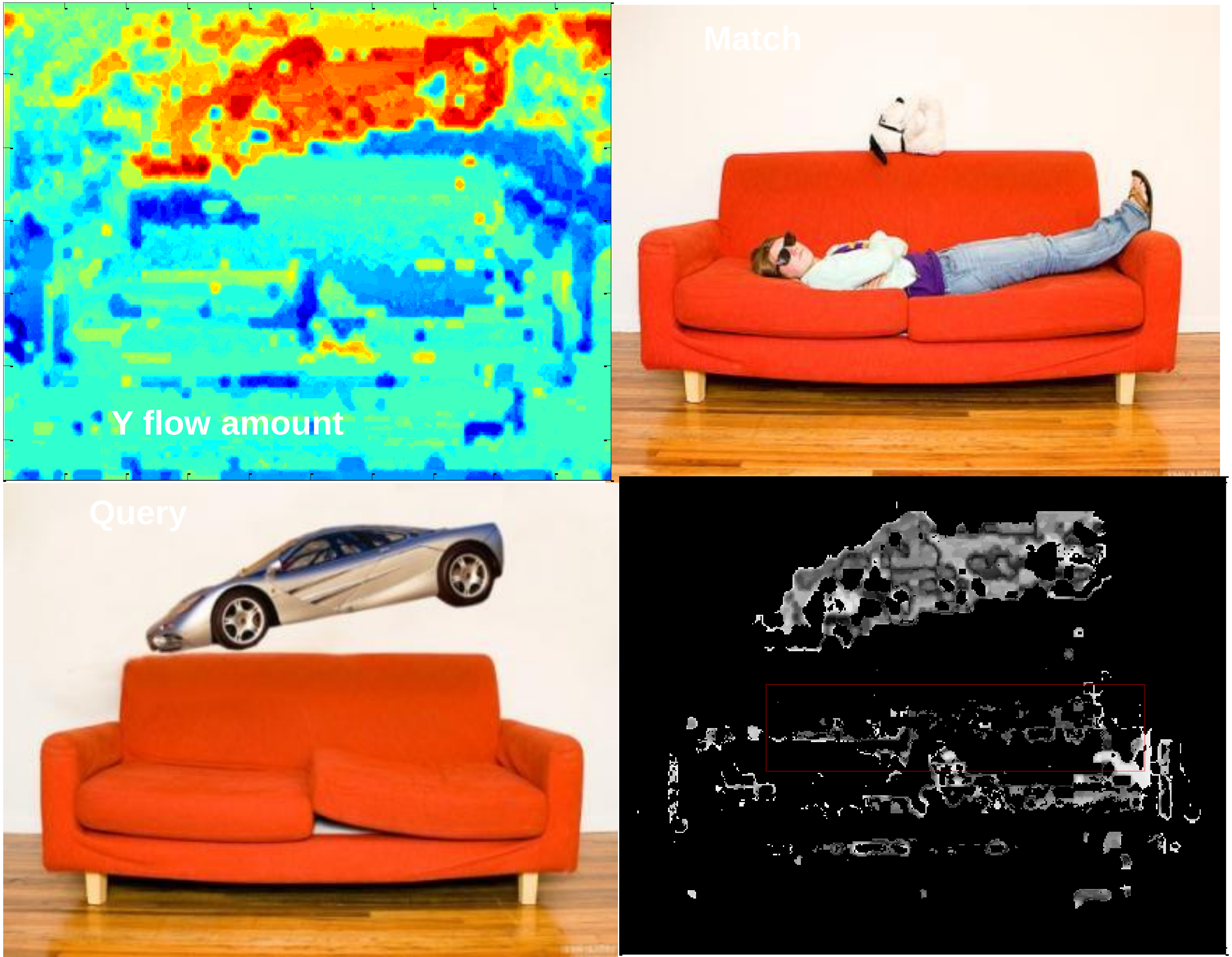


SIFT Flow Hard Example

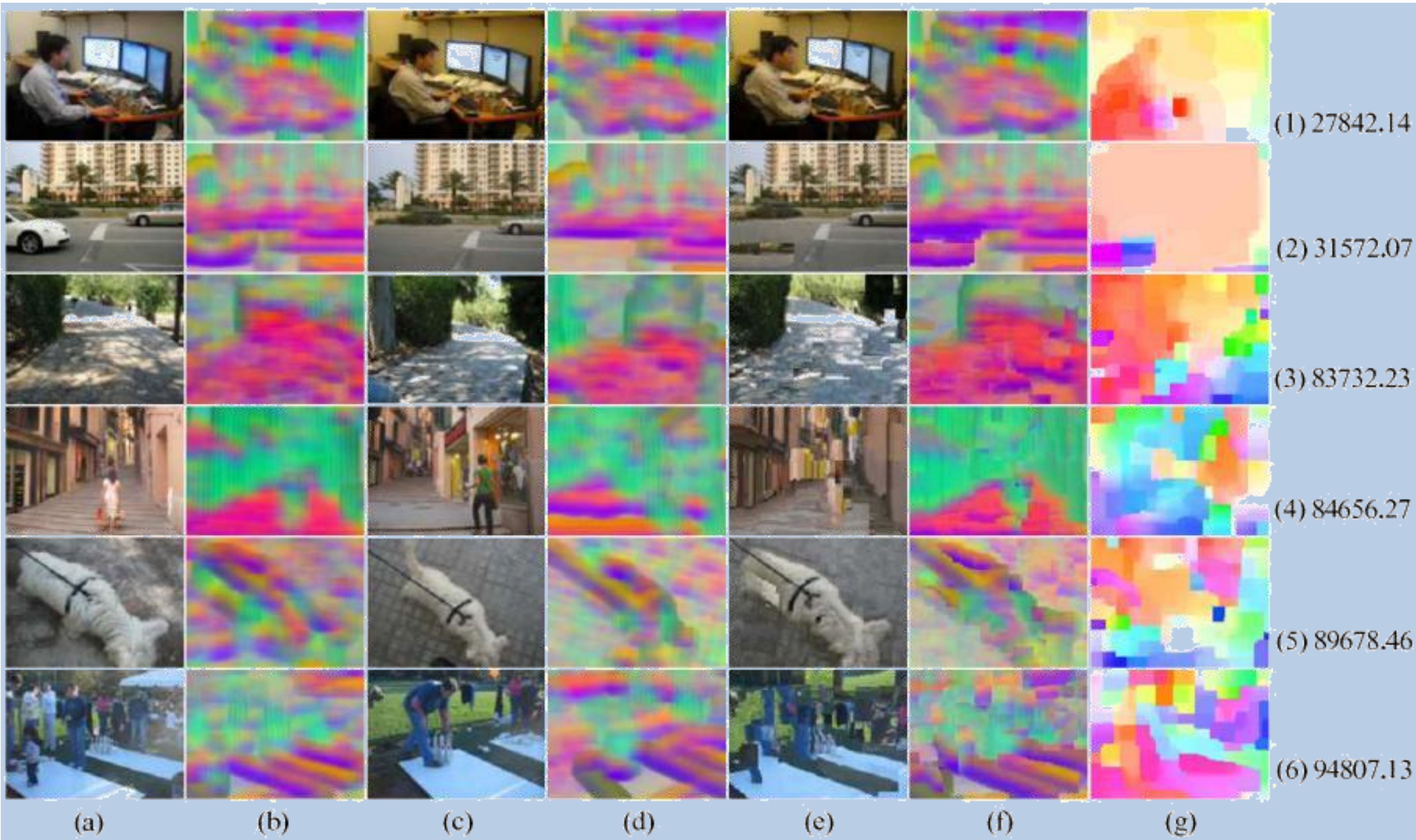


SIFT Flow Hard Example

SIFT Flow Hard Example



SIFT Flow Paper Examples



SIFT Flow Paper Examples

Estimating Motion

- What else can we do with SIFT Flow?

Original Image | **Database Match** | **Motion of Database Match** | **Warped and Transferred Motion** | **Ground Truth of Original Image**

Motion Ambiguity

- Multiple plausible motions

Synthesizing Motion

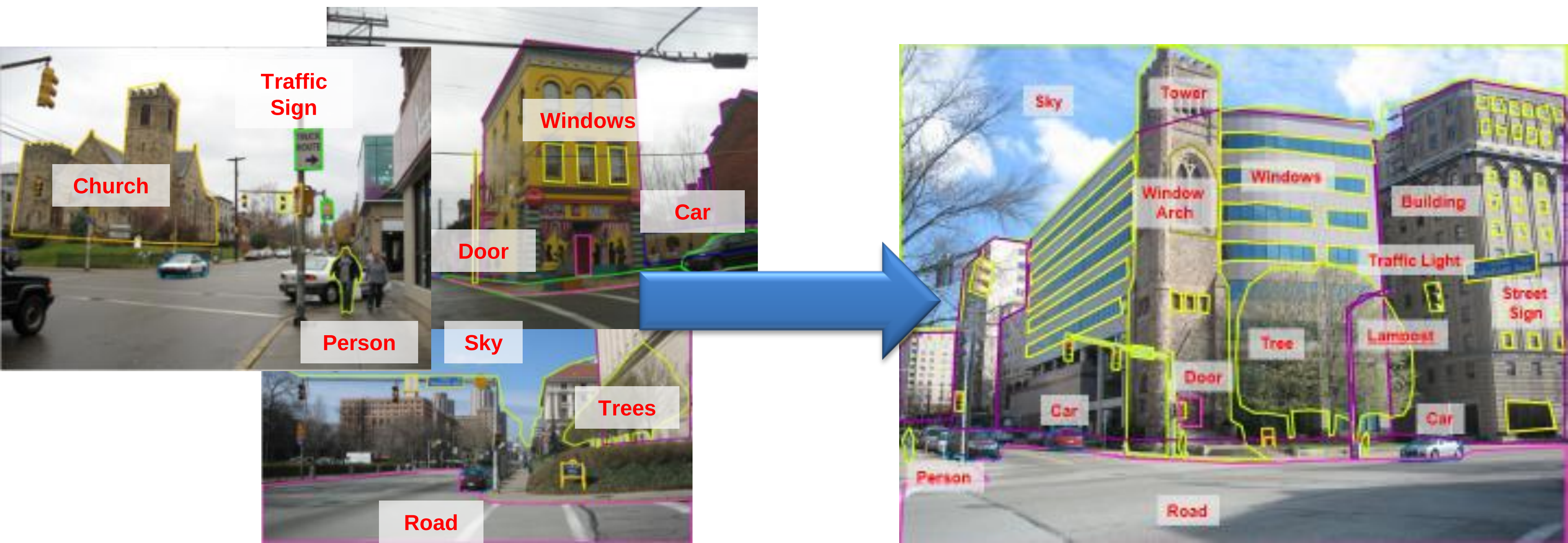
Input Image **Composite Video** **Retrieved Motion**

Papers

- SIFT Flow isn't quite there yet
- If you can't match objects in images
  - Find similar, but non-spatially aligned scenes
  - Use labeled information as a prior
- Context Paper
  - B. C. Russell, A. Torralba, C. Liu, R. Fergus, W.T. Freeman. "Object Recognition by Scene Alignment." NIPS 2007.

Object Detection

- Use a “context-enhanced” sliding window
- Retrieve K similar scenes and extract priors
  - Frequency and spatial information
  - Weaker form of label transfer based on “clues”



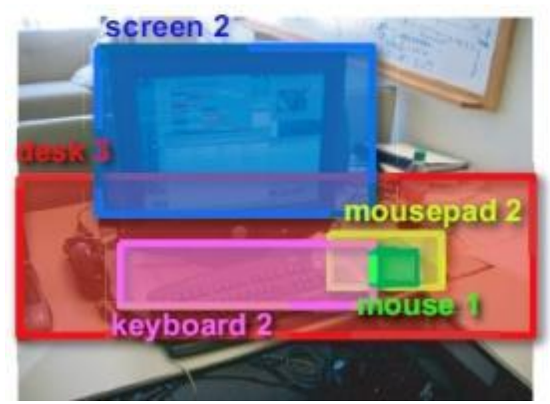
Context Approach

- Goal: Recognize objects embedded in a scene

Input image

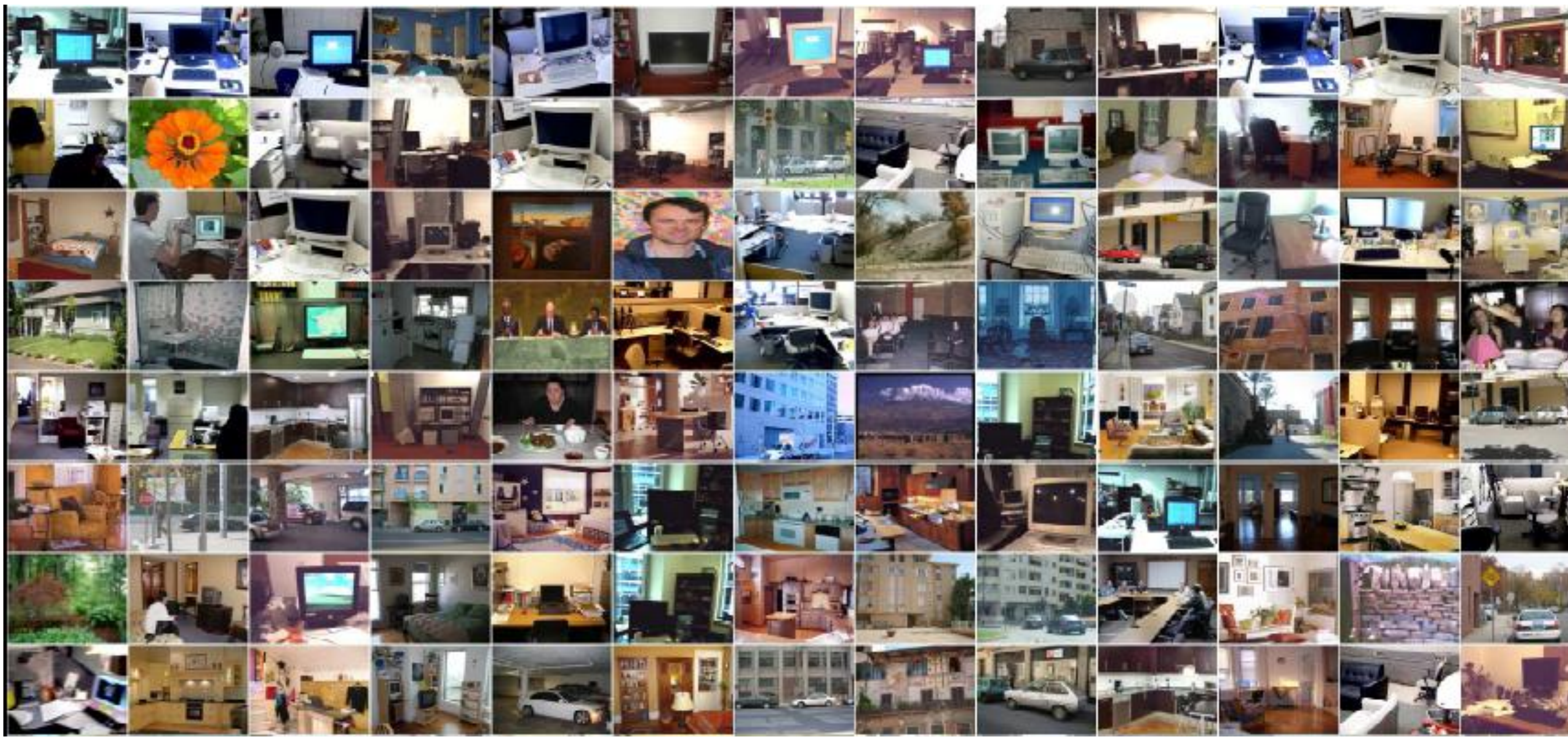
Nearest neighbors from 15,691 images

Cluster images using object labels

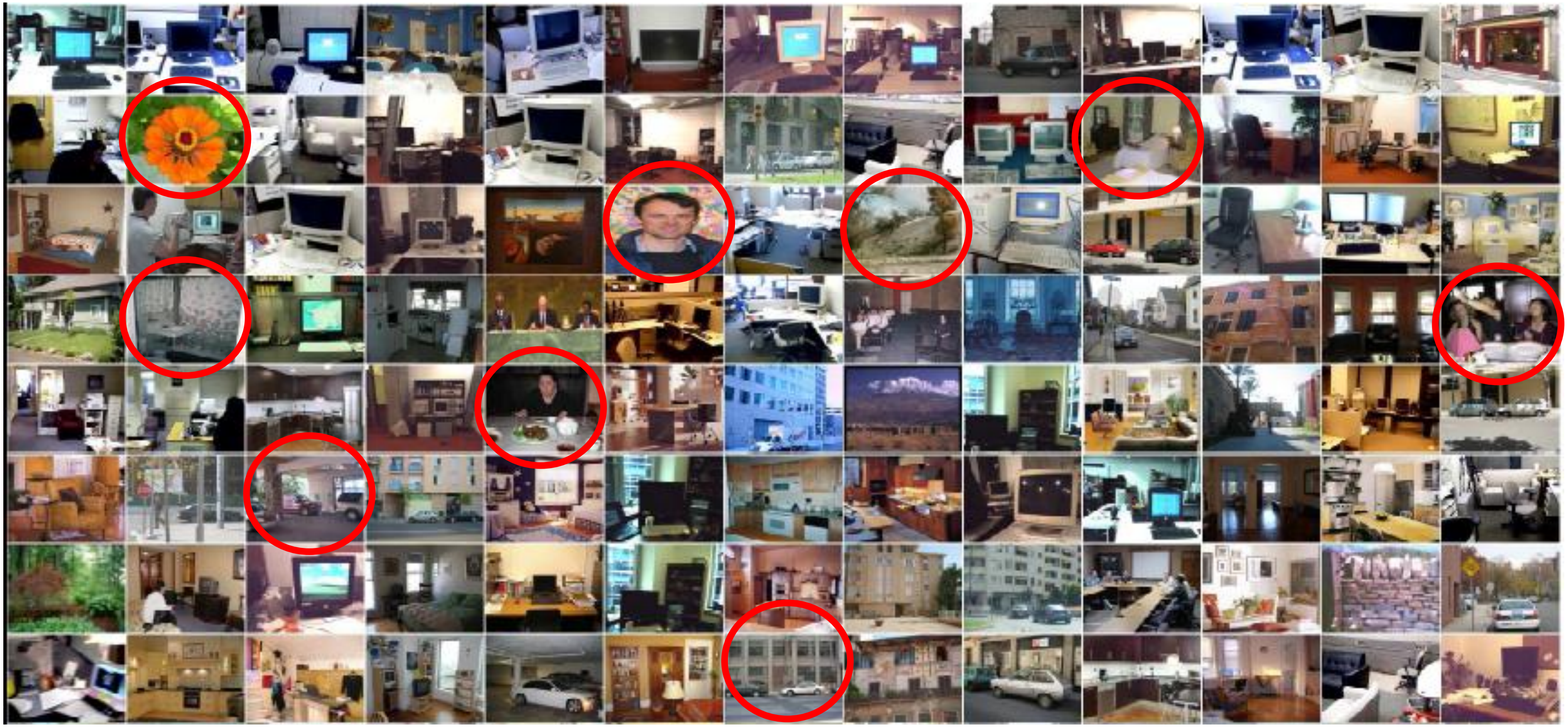


Output image with object labels transferred

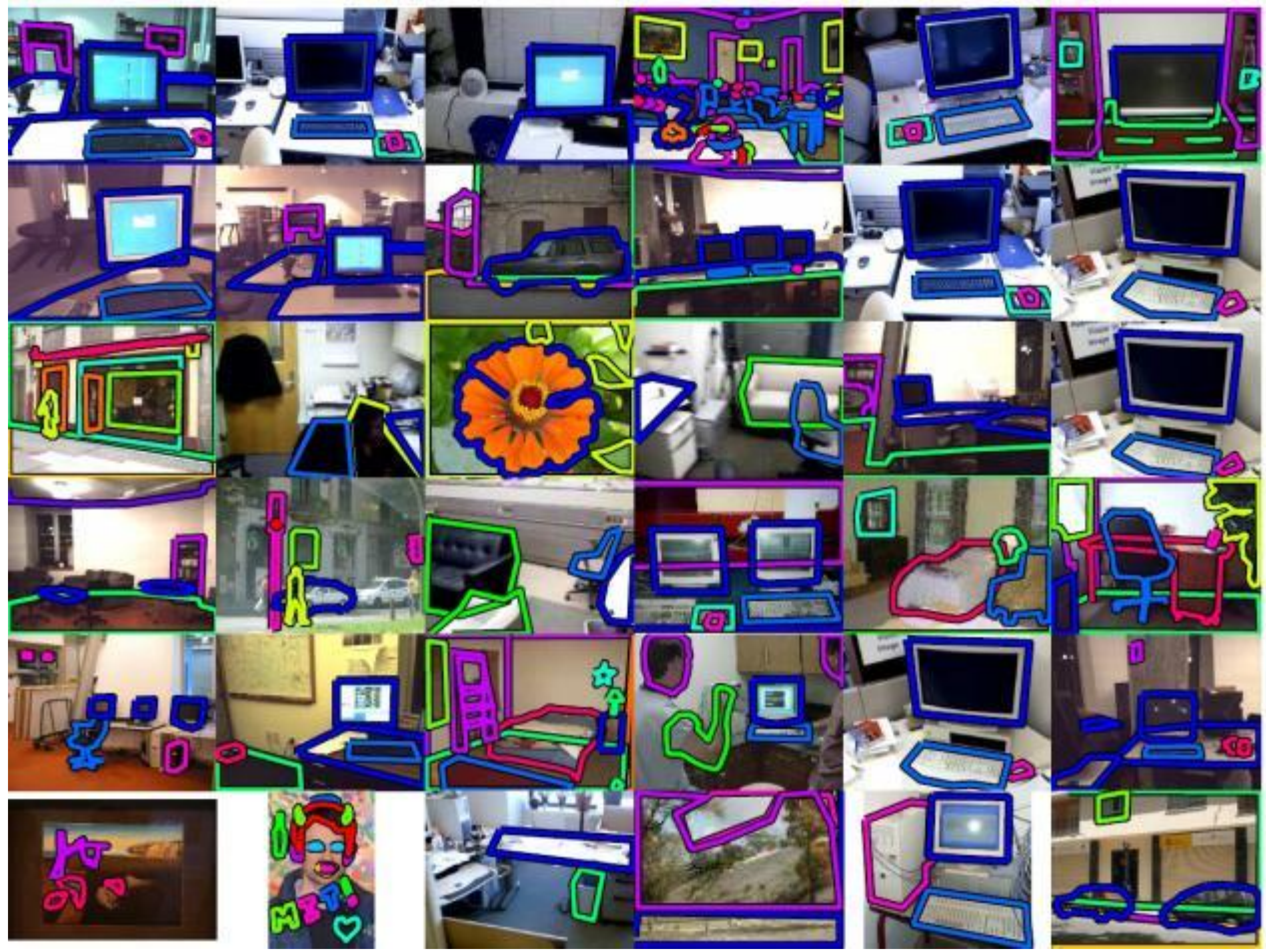

JAVA




JAVA



Retrieval set + LabelMe labels



- Steal object
  - Frequency
  - Location
  - Size
  - Etc

Goals

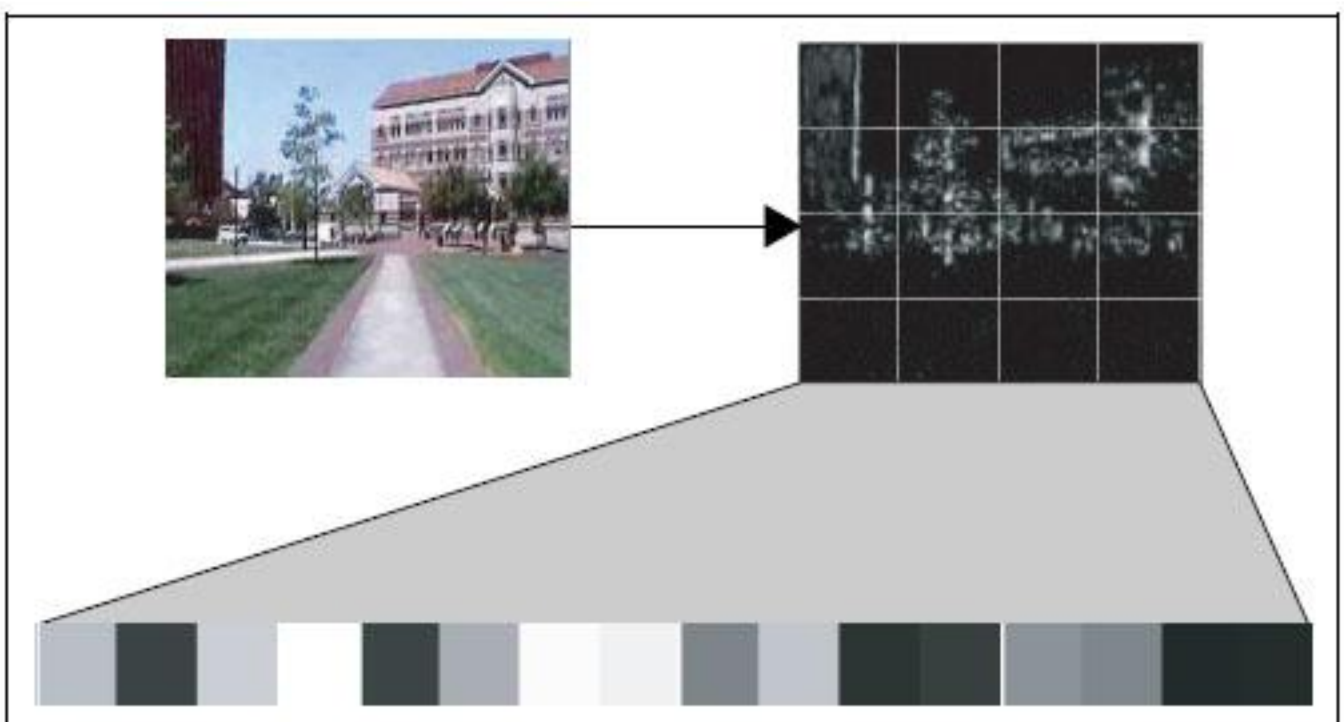
- Given *db*: A database of labeled images
- Given *img*: A new image

- Find images similar to *img* in *db*
  - Similar scenes (mountain, office, etc)
  - Similar objects (coffee cup, car, etc)
  - Similar layout (lake on left, building to right)

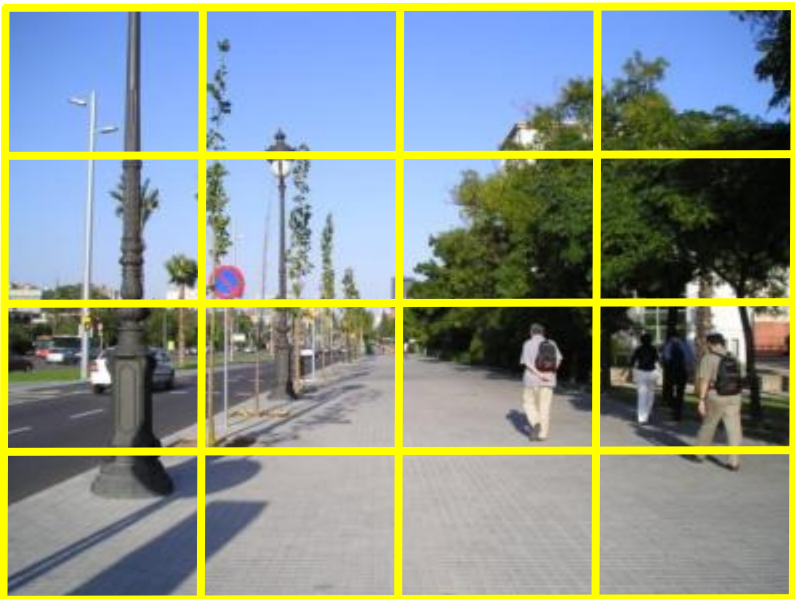
- Basically, scene alignment

Matching Gist Features

- Decompose image into scene descriptors
- Gist features (A. Oliva, et. al. 2001)



Matching Gist Features



- Apply oriented Gabor filters over different scales
- Average filter energy in each bin

8	orientations
4	scales
x 16	bins
512	dimensions

- Used for scene recognition
- Similar to SIFT (Lowe 1999)

Evaluation Dataset

- Used a subset of the Labelme dataset

- Training:
  - 15,691 images
  - 105,034 labels

- Testing:
  - Cities/offices outside of training set
  - 560 images

Predicting Object Presence

- Can descriptor predict the presence of

→ Descriptor →

Does this image contain:
- Person?
- Computer monitor?
- Building?
- Beer?
- Car?
- Etc…

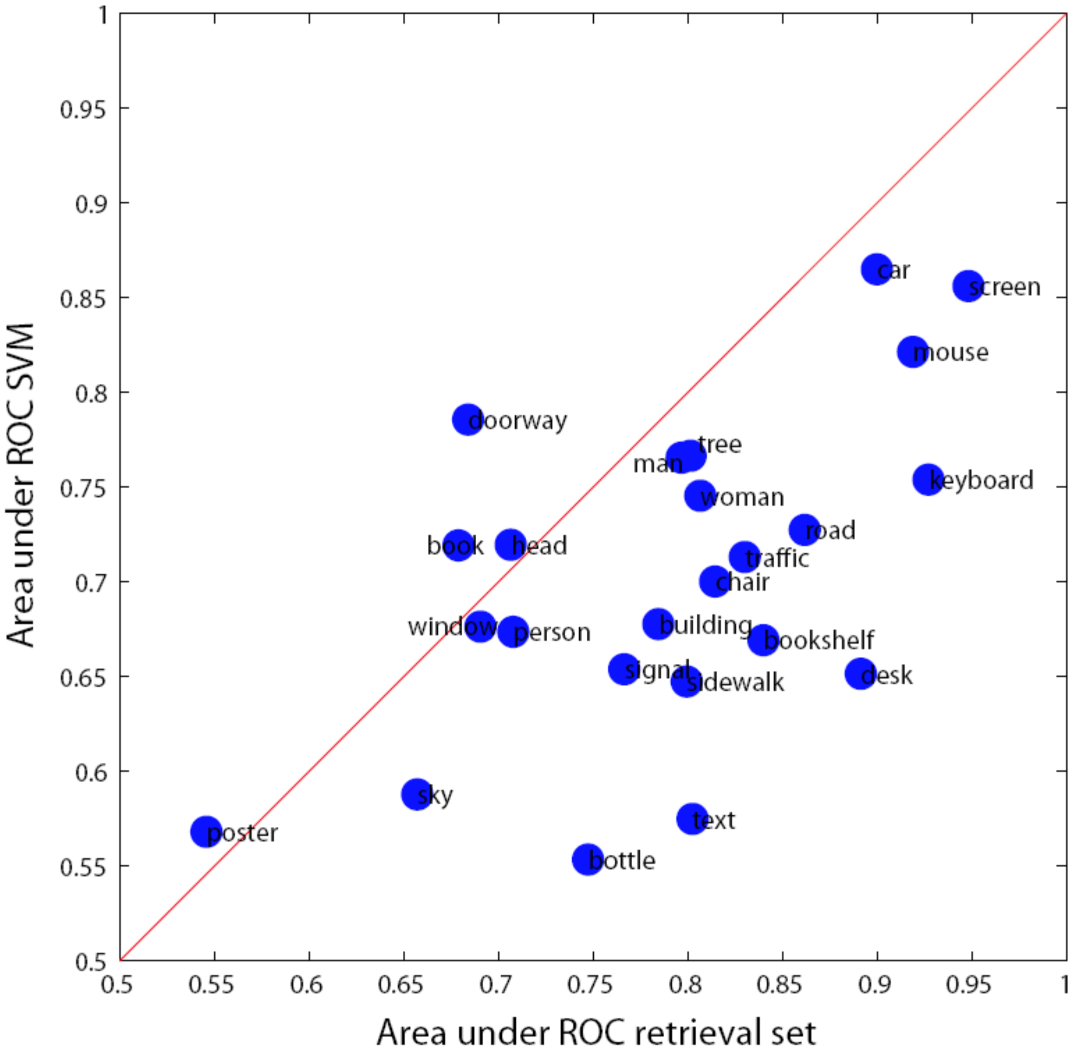
- Or use indirect method of matching images

→ Descriptor →

→

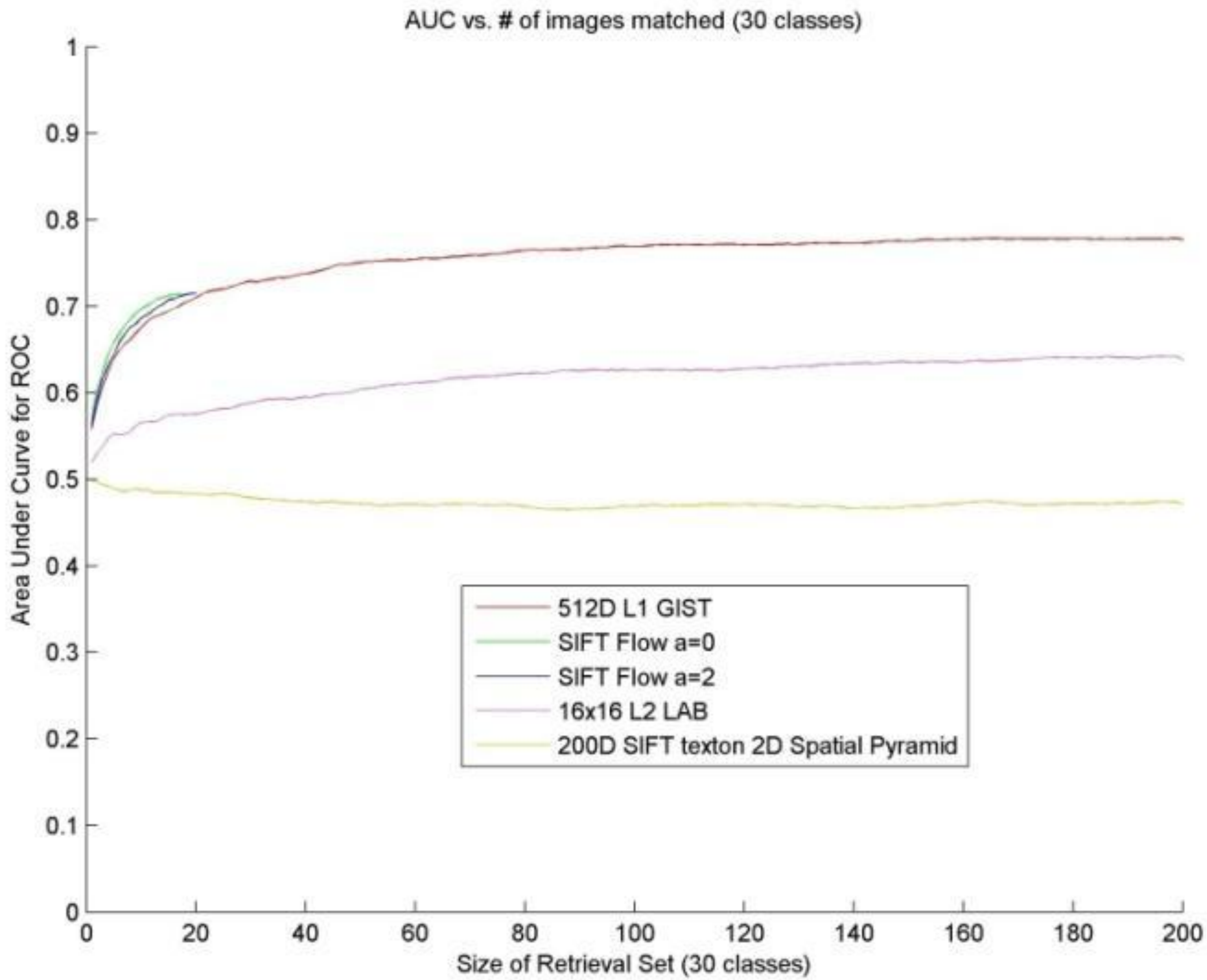
Do these images contain:
- Person?
- Computer monitor?
- Building?
- Beer?
- Car?
- Etc…

SVM Object vs. kNN

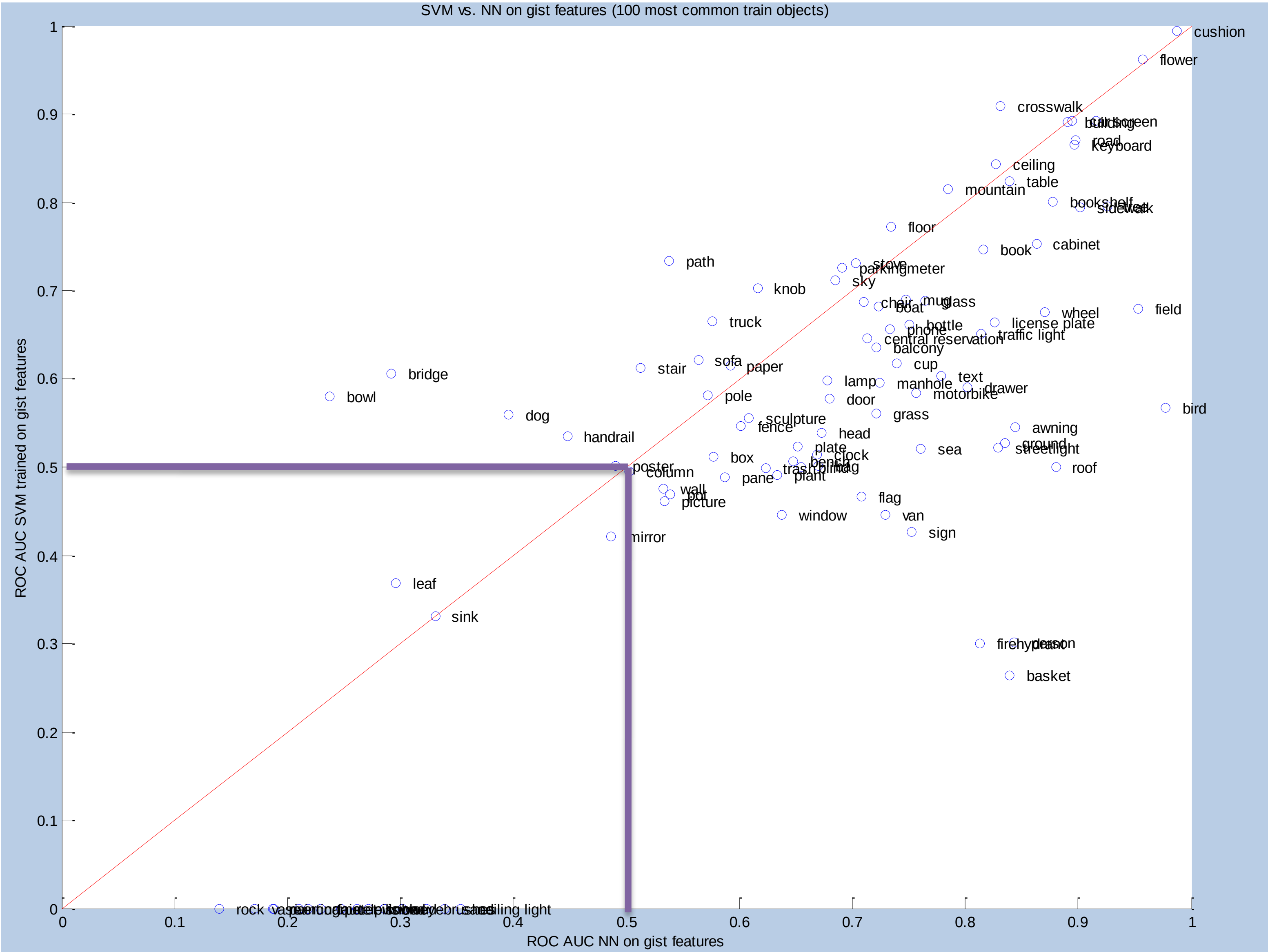


- Per object SVM
  - SVM trained on object bounding box gist features
  - SVM applied to bounding boxes in image
  - Maximal score used
- Retrieval set:
  - Histogram object labels
  - Use normalized histogram value to classify image

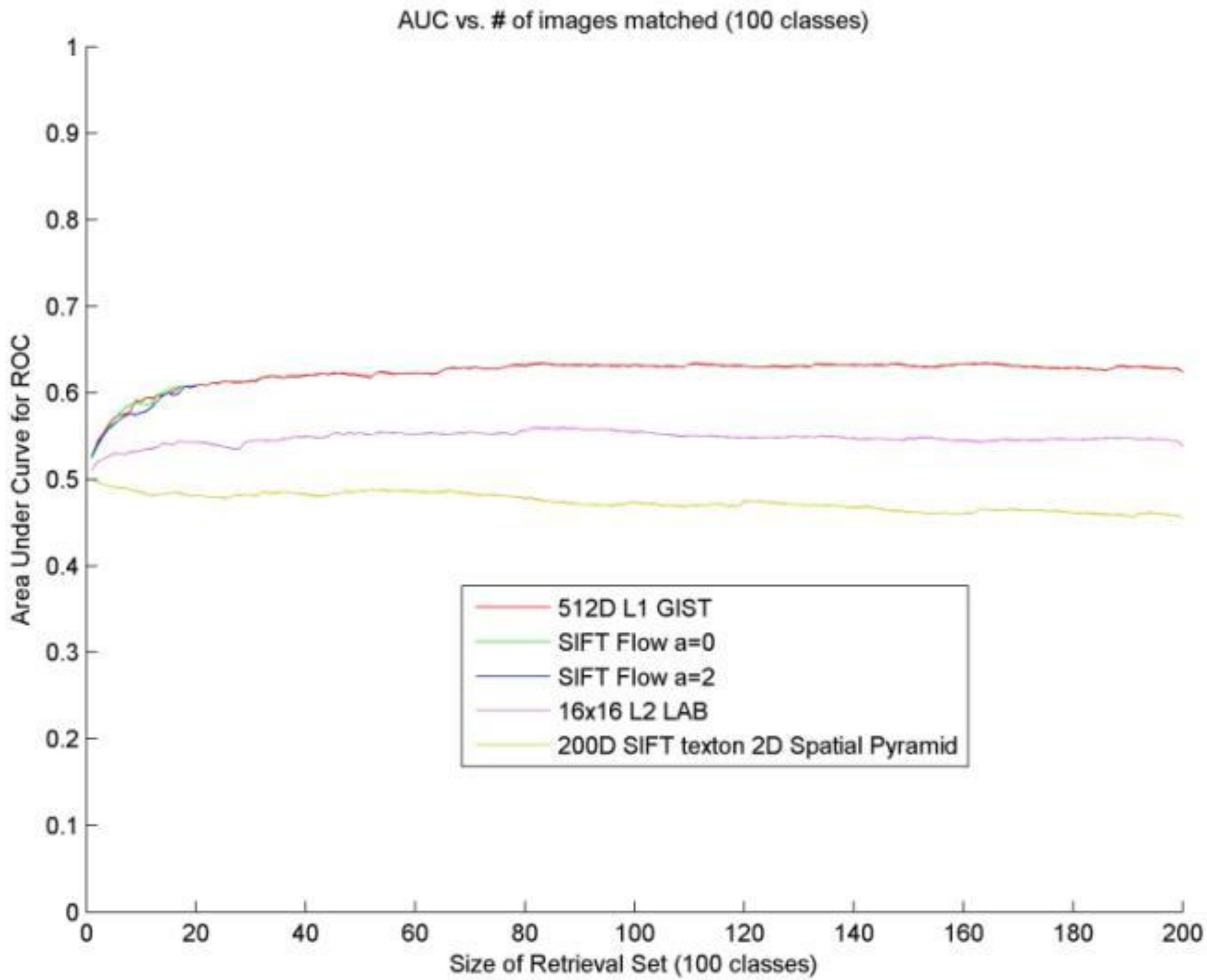
Method/k Comparison



SVM (image) vs. kNN



Method/k Comparison



Using Retrieval Set

- Object detection uses variable-sized sliding windows and an SVM appearance model
  - Very slow, ~4,000 bboxes to calculate gist for
- Find contextual clues in retrieval set
  - If all the matched images were of streets, unlikely to find a keyboard
- Build a probabilistic model including information transferred from matched images

Using Retrieval Set

- Probabilistic Formulation

The likelihood of objects appearing in the image

Object Spatial Info

The likely spatial locations of observing object class *l* in the image

Object Classes

Object Appearance

The appearance likelihood of object category *l*

$$p(o, x, g | \theta, \phi, \eta) = \prod_{i=1}^{N} \prod_{j=1}^{M_i} \sum_{h_{i,j}=0}^{1} p(o_{i,j} | h_{i,j}, \theta)\, p(x_{i,j} | o_{i,j}, h_{i,j}, \phi)\, p(g_{i,j} | o_{i,j}, h_{i,j}, \eta)$$

  - N images, M object proposals per image, L classes
  - $h_{i,j}=1$ indicates object class $o_{i,j}$ is present at location $x_{i,j}$

Using Retrieval Set

- Probabilistic Formulation

$$p(o,x,g|\theta,\phi,\eta) = \prod_{i=1}^{N}\prod_{j=1}^{M_i}\sum_{h_{i,j}=0}^{1} p(o_{i,j}|h_{i,j},\theta)\, p(x_{i,j}|o_{i,j},h_{i,j},\phi)\, p(g_{i,j}|o_{i,j},h_{i,j},\eta)$$

  - Spatial locations encoded by centroid & size of bounding box of object (normalized to [0,1])

$$x_{i,j} = (c^x_{i,j}, c^y_{i,j}, c^w_{i,j}, c^h_{i,j})$$

  - Probability parameters θ_m and $\phi_{m,l}$ are learned from the retreival set on
  - Probability parameter $\eta_{m,l}$ is learned offline by training an SVM for each object class on training set

Using Retrieval Set

- Advantages
  - Can increase accuracy if retrieval set is good
  - Can save CPU time by constraining search
    - Look only for objects likely to be in the image
    - Look only for objects in likely locations
- Disadvantages
  - Can decrease accuracy if retrieval set is bad
  - Non-exhaustive search can miss objects
    - Maybe there is a bike indoors

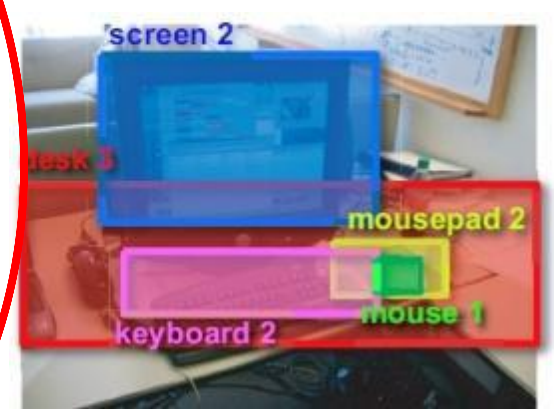
Context Approach

- Goal: Recognize objects embedded in a scene

Input image

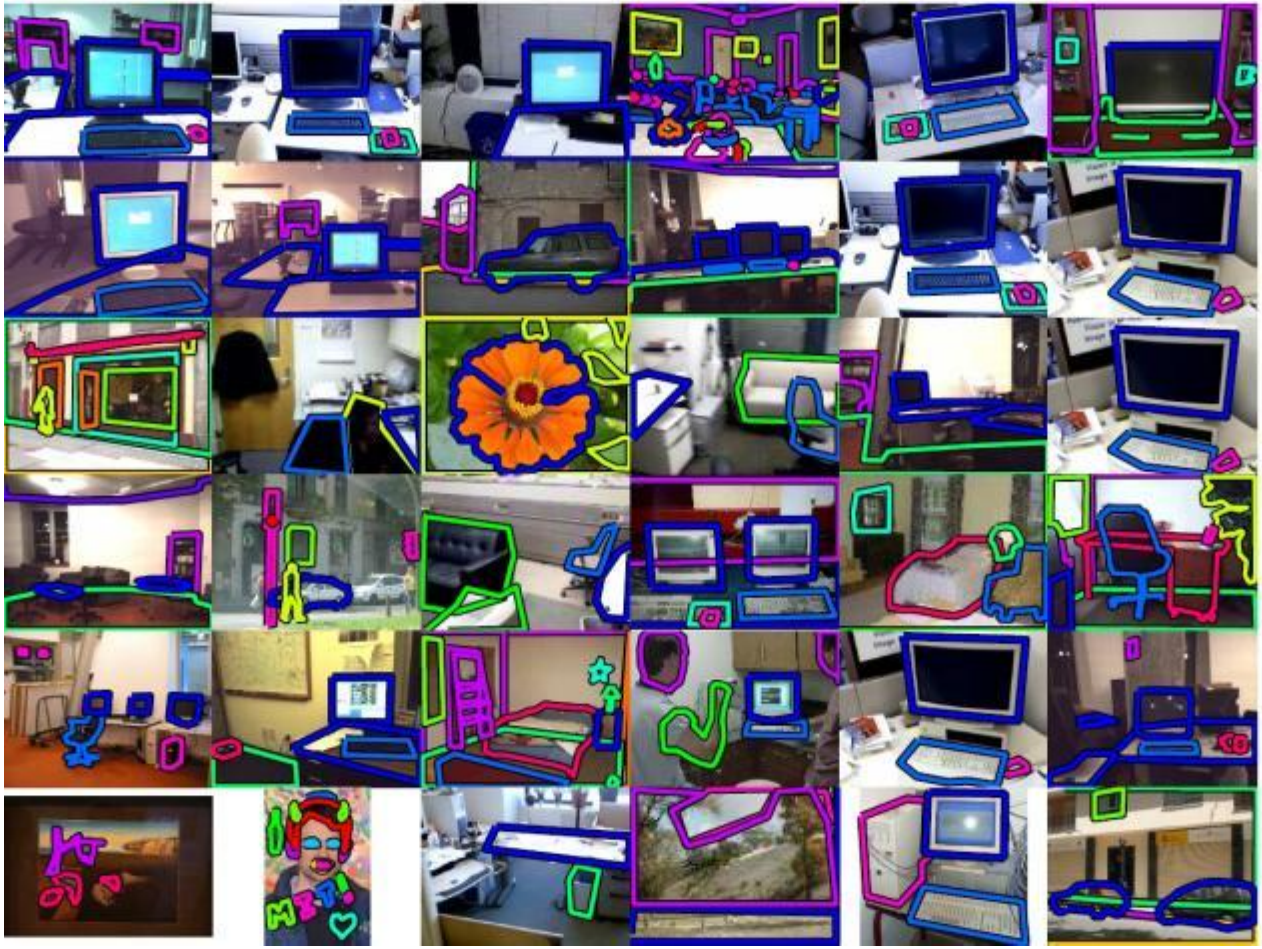
Nearest neighbors from 15,691 images

Cluster images using object labels



Output image with object labels transferred

Clustering Retrieval Set



Cluster images based on labels:

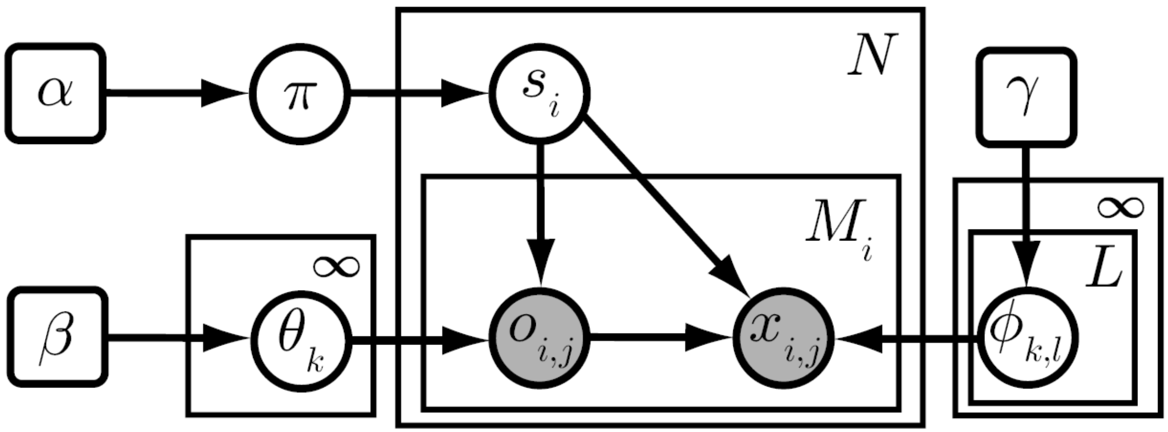
- Object identity
- Location within image

Clustering Retrieval Set

- “Used a simple model to cluster object labels belonging to the retrieved images”

- Incorporate latent clusters with mixing weights
- Cluster object labels and spatial locations
- Dirichlet process prior with stick-breaking
- Rao-Blackwellized Gibbs sampler
- Manually tuned hyperparameters
- Perform hard Expectation Maximization (EM)

Clustering Retrieval Set



s_i - cluster assignment

o_{ij} - object labels

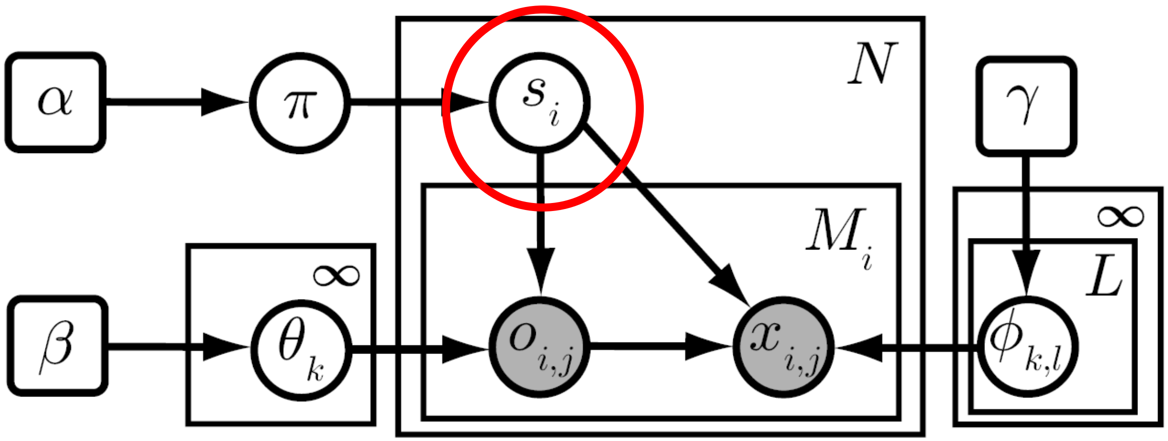
x_{ij} - bounding box parameters

$$s_i|\pi \sim \pi \qquad \pi|\alpha \sim Stick(\alpha)$$

$$o_{i,j}|s_i = k, \theta \sim \theta_k \qquad \theta_k|\beta \sim Dirichlet(\beta)$$

$$x_{i,j}|s_i = k, o_{i,j} = l, \phi \sim \mathcal{N}(\phi_{k,l}) \qquad \phi_{k,l}|\gamma \sim \mathcal{NIW}(\gamma)$$

Clustering Retrieval Set



s_i - scene assignment

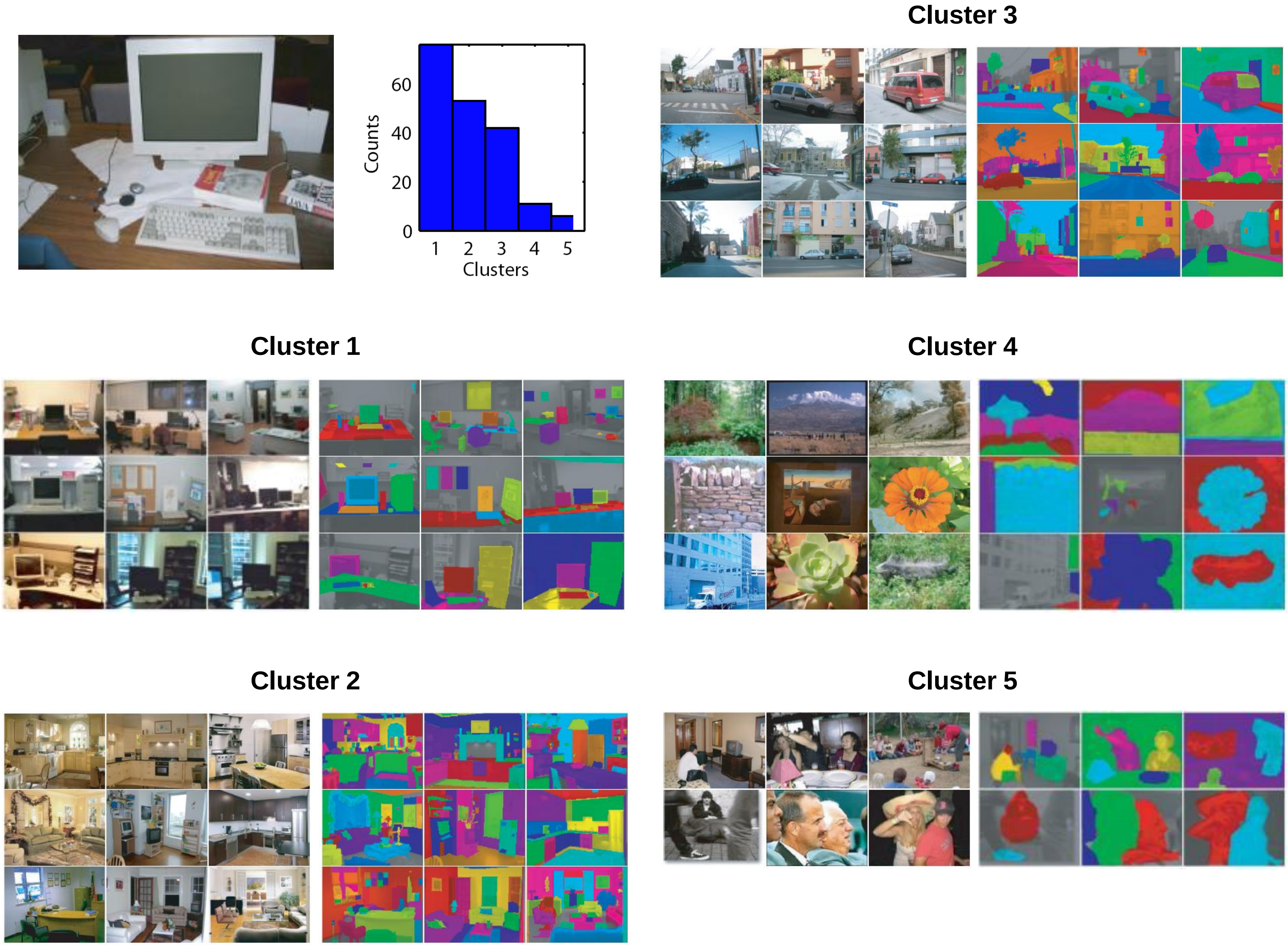
o_{ij} - object labels

x_{ij} - bounding box parameters

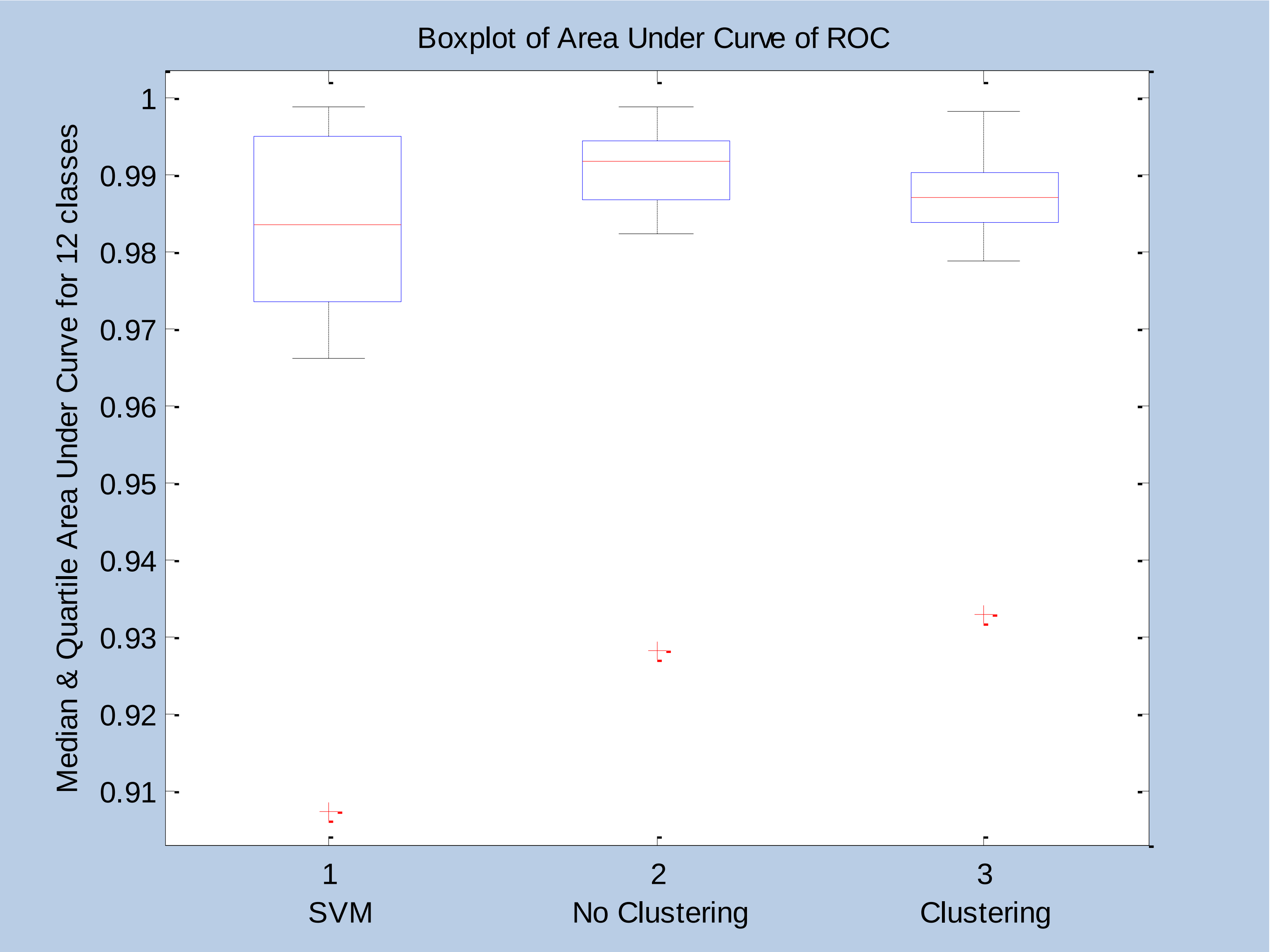
Use Gibbs sampler to draw scene assignments:

$$s_i \sim p(s_i | s_{\setminus i}, o, x, \alpha, \beta, \gamma)$$

Chinese restaurant process analogy:
tables - scene parameters; customers - images


Counts
60
40
20
0
1
2
3
4
5
Clusters
Cluster 3
Cluster 1
Cluster 4
Cluster 2
Cluster 5

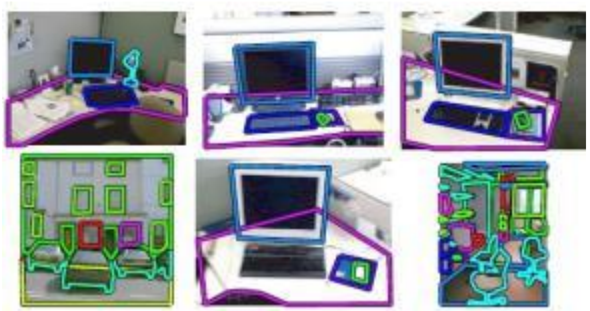
Results: ROC Curves



Context Approach

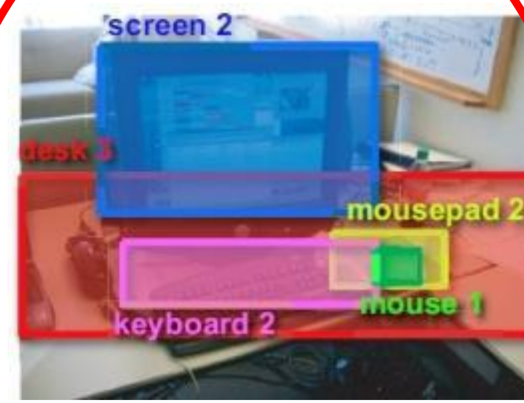
- Goal: Recognize objects embedded in a scene

Input image



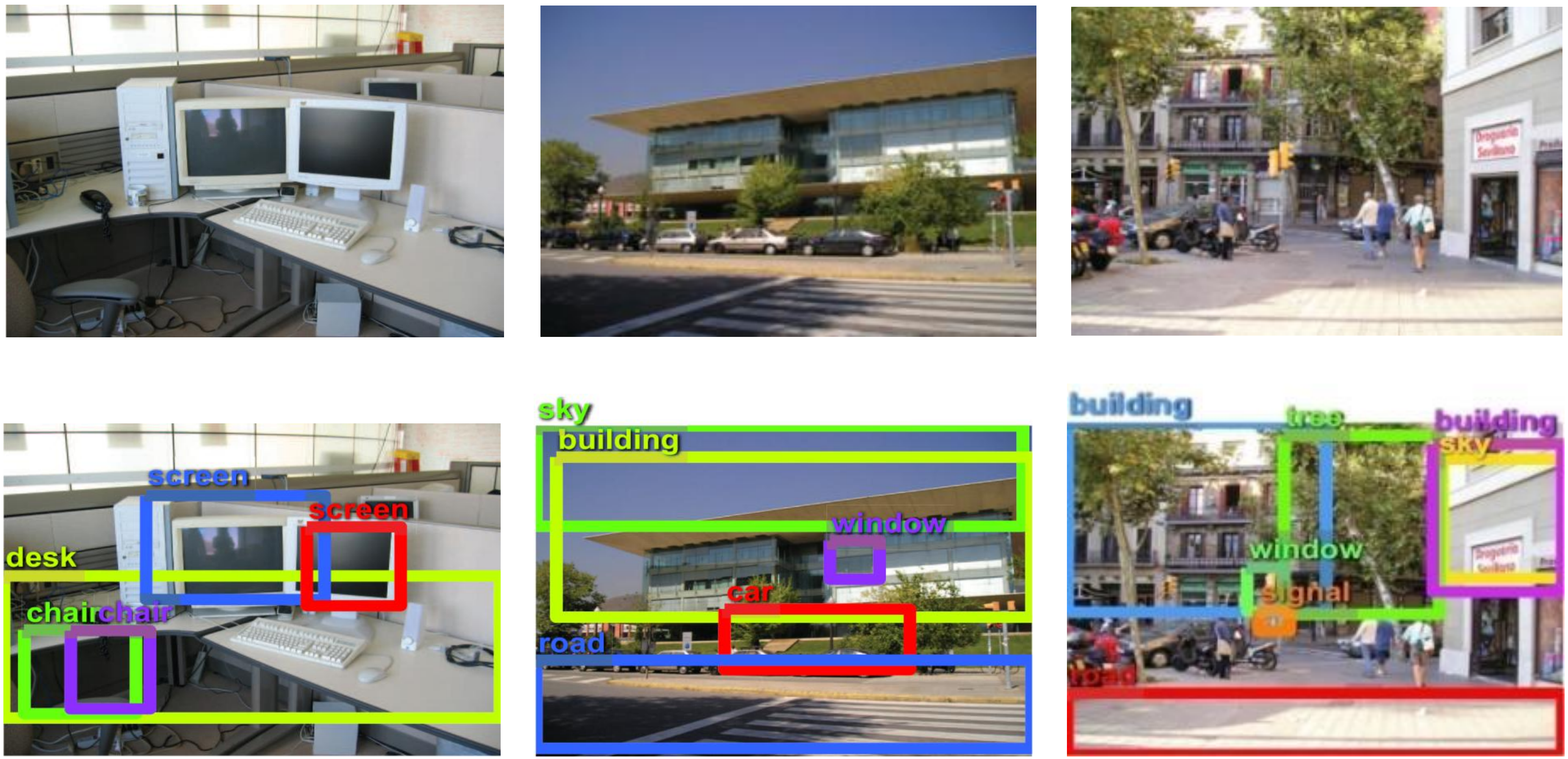
Nearest neighbors from 15,691 images

Cluster images using object labels



Output image with object labels transferred

Outputs

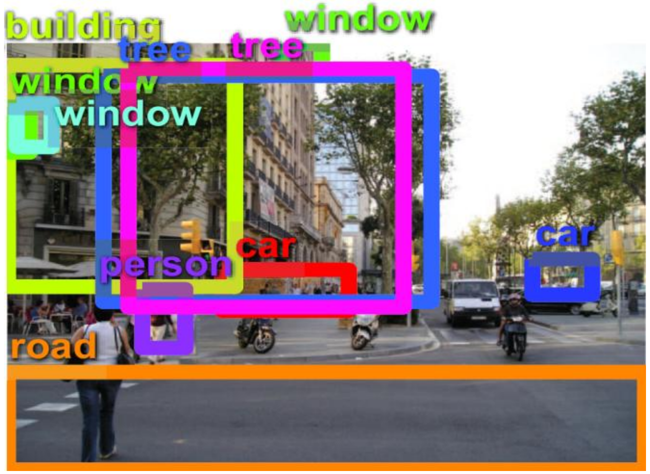


screen
screen
desk
chair
chair
sky
building
window
car
road
building
tree
building
sky
window
signal
road

Outputs

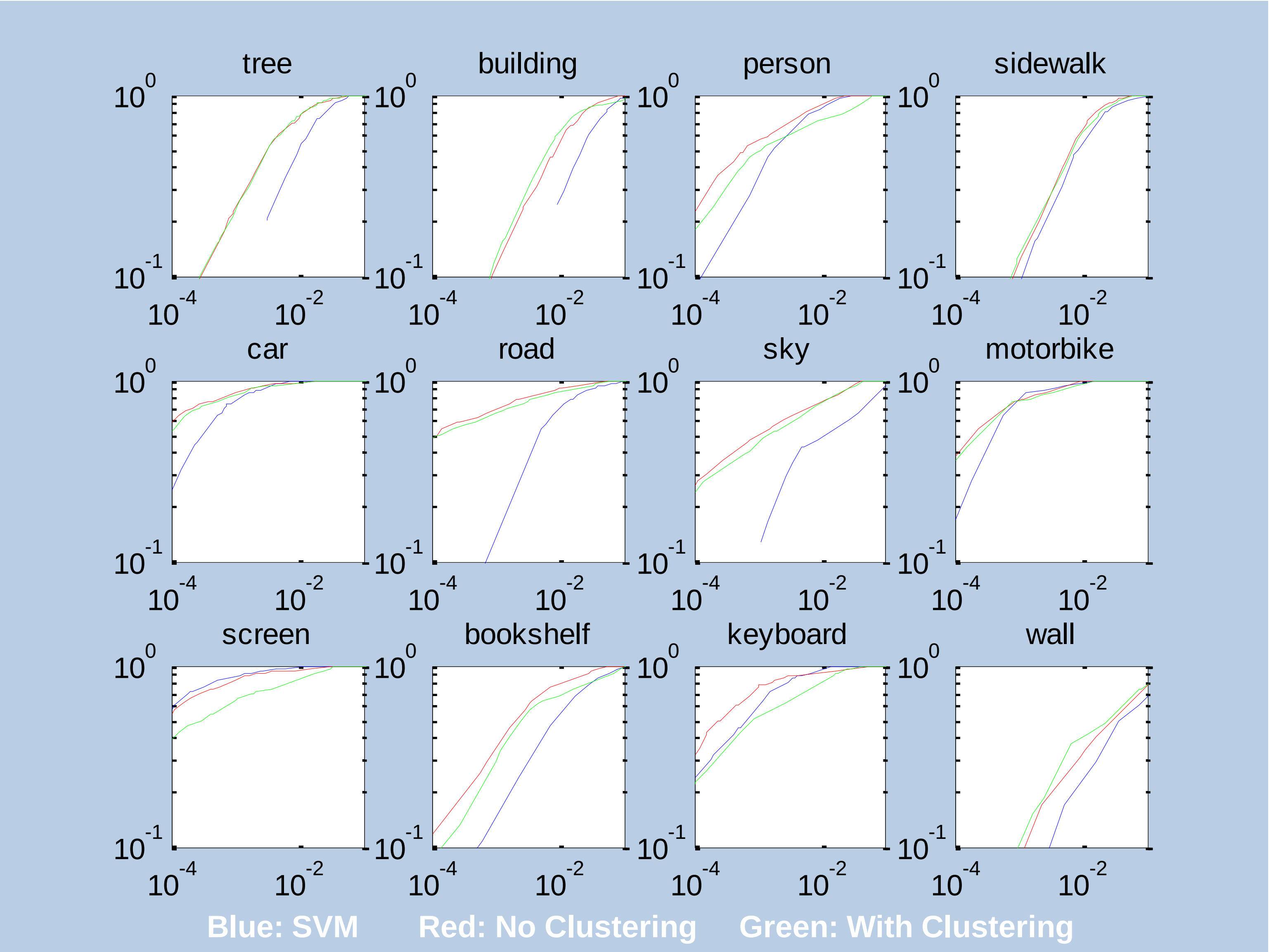
ARRIGA
MUEBLES

building
sky
tree
window
window
window
signal
car
car
road
sidewalk
ARRIGA
MUEBLES

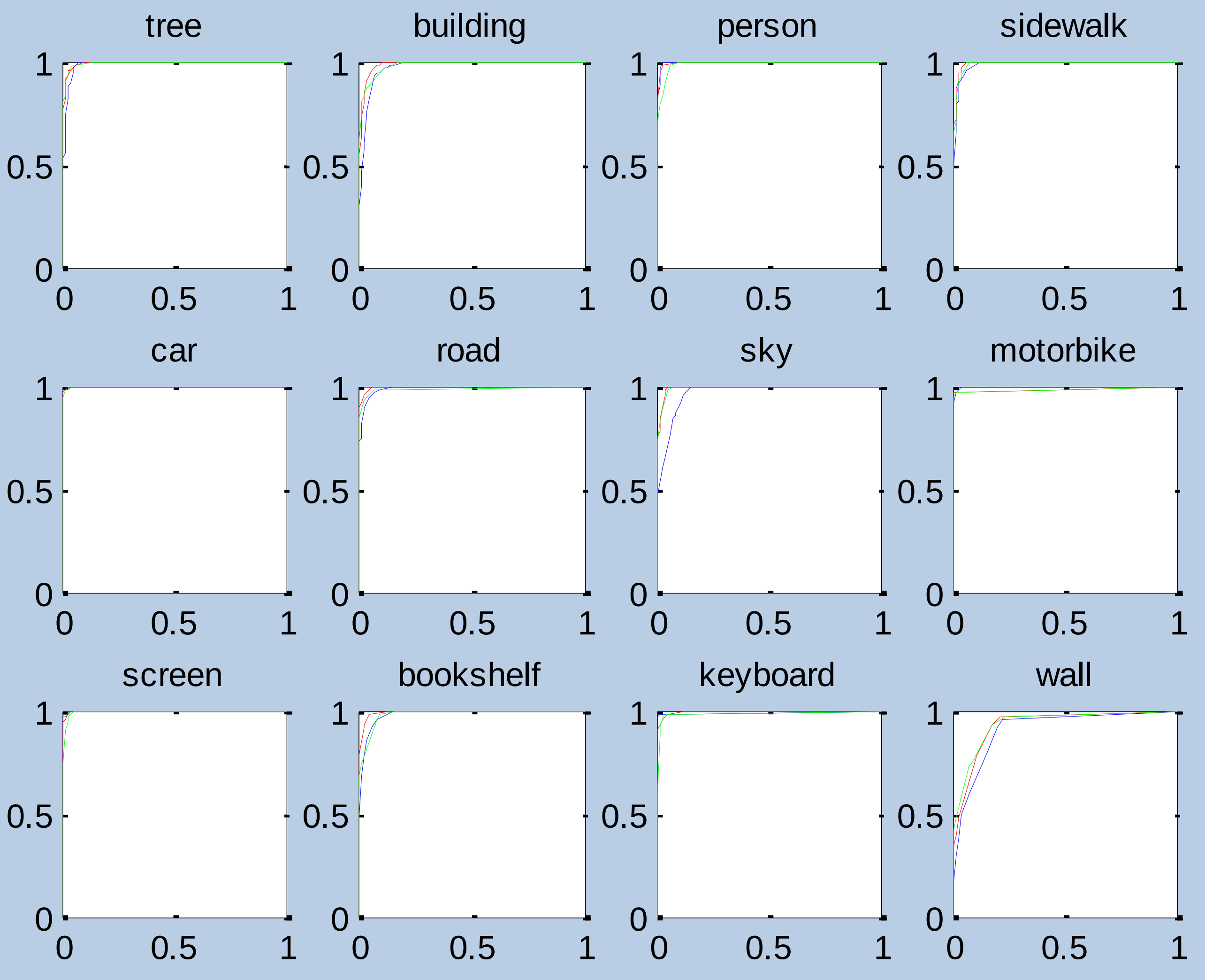


building
tree
tree
window
window
person
car
car
road

Results: ROC Curves

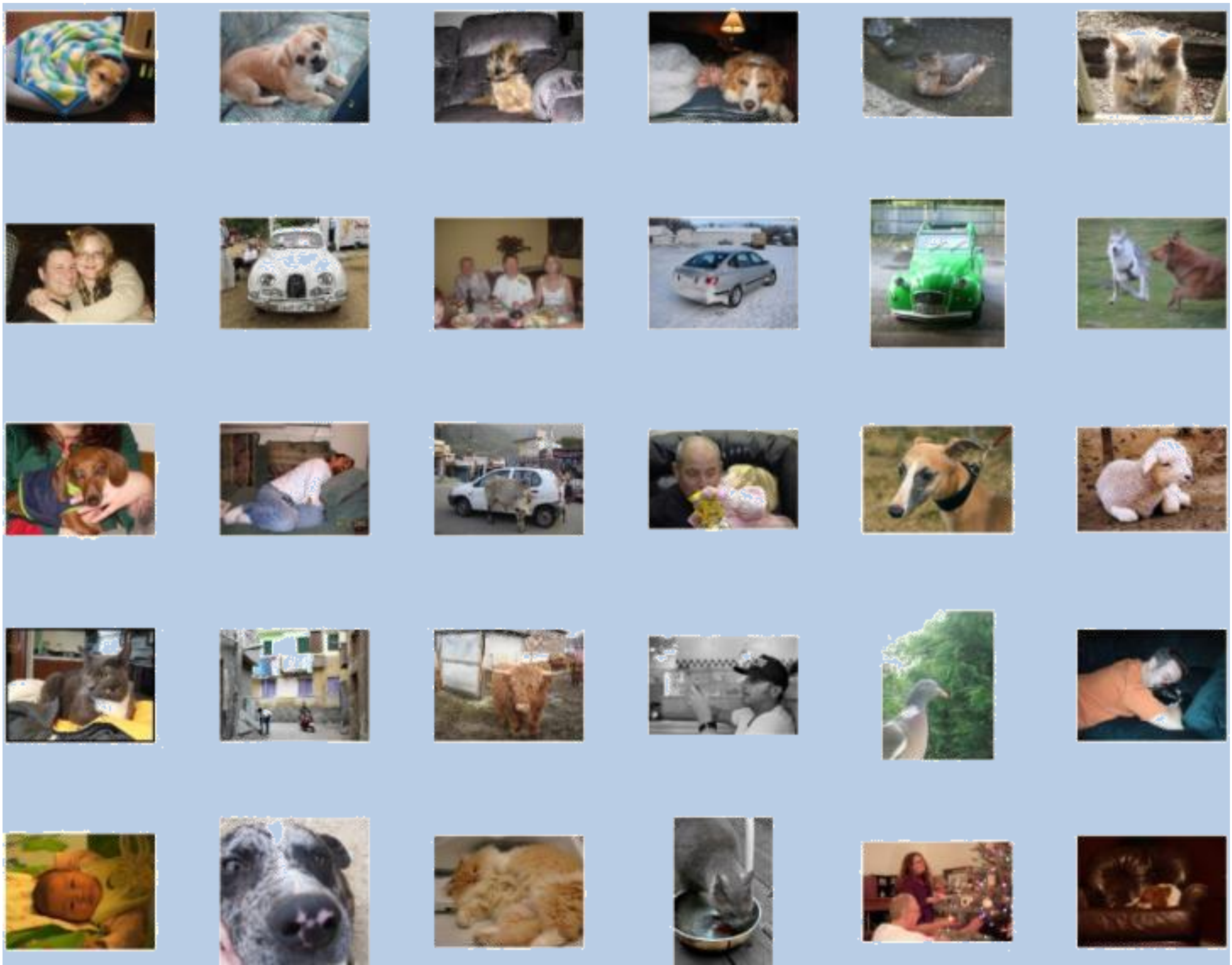


Results: ROC Curves

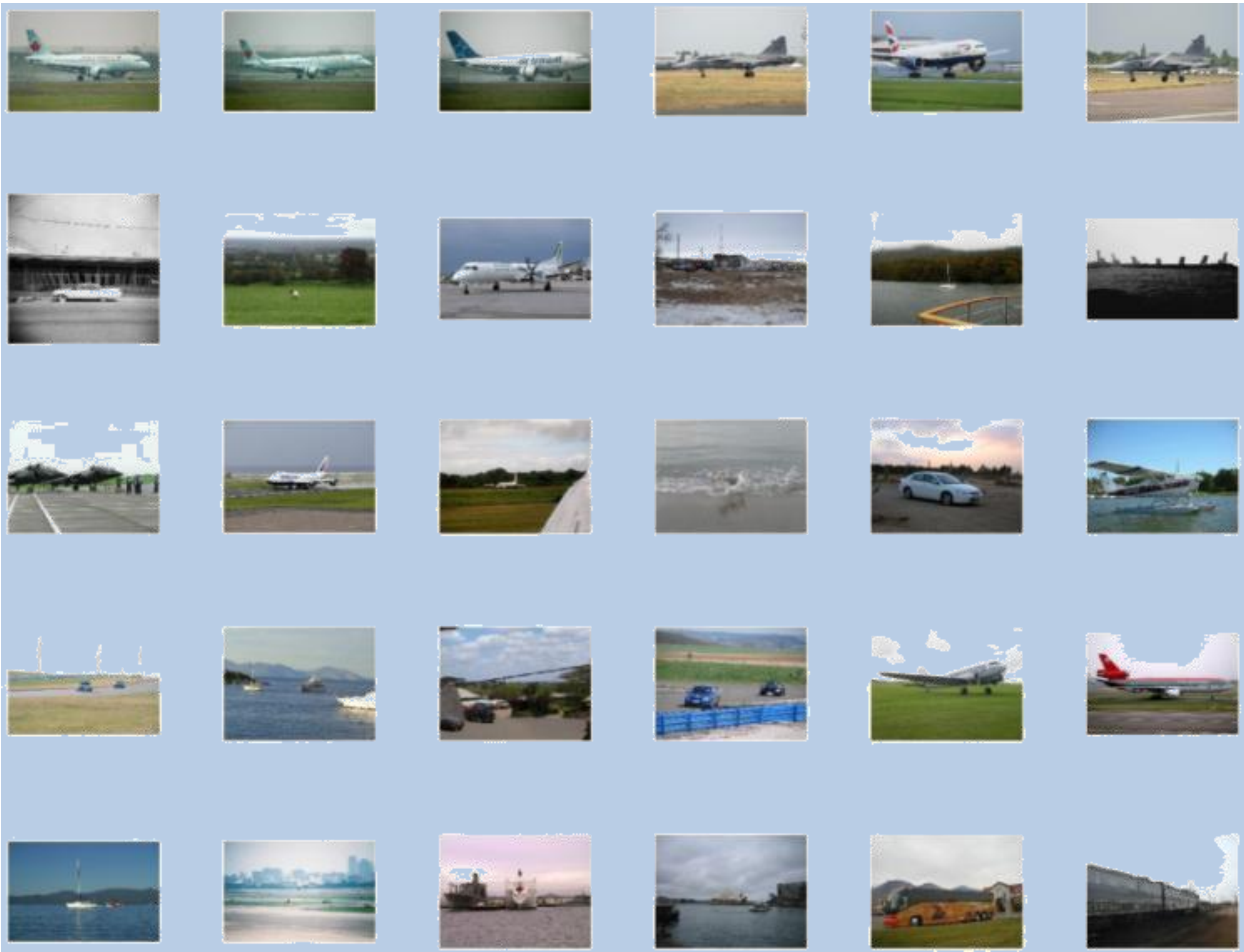


Blue: SVM Red: No Clustering Green: With Clustering

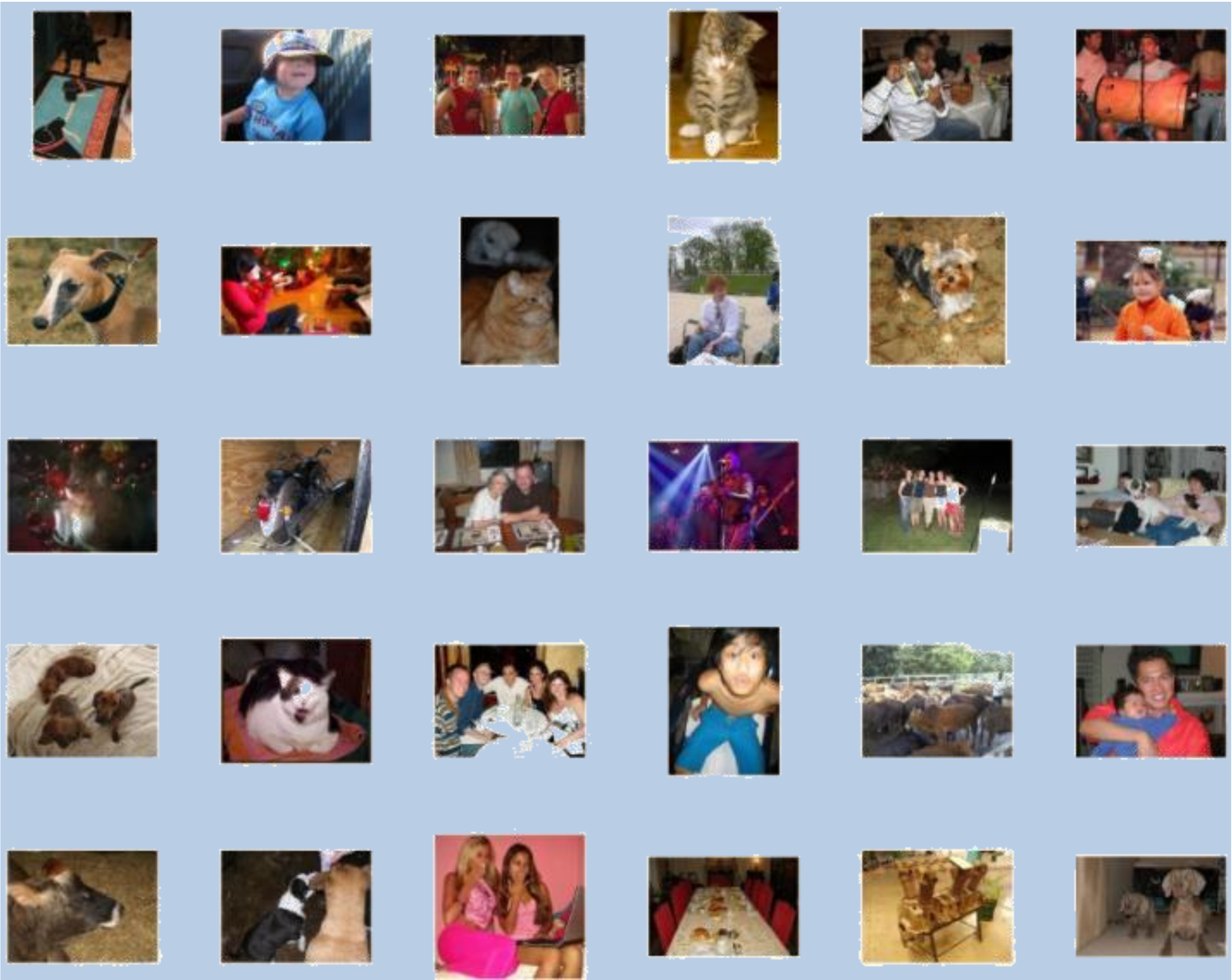
Pascal 2007 Results



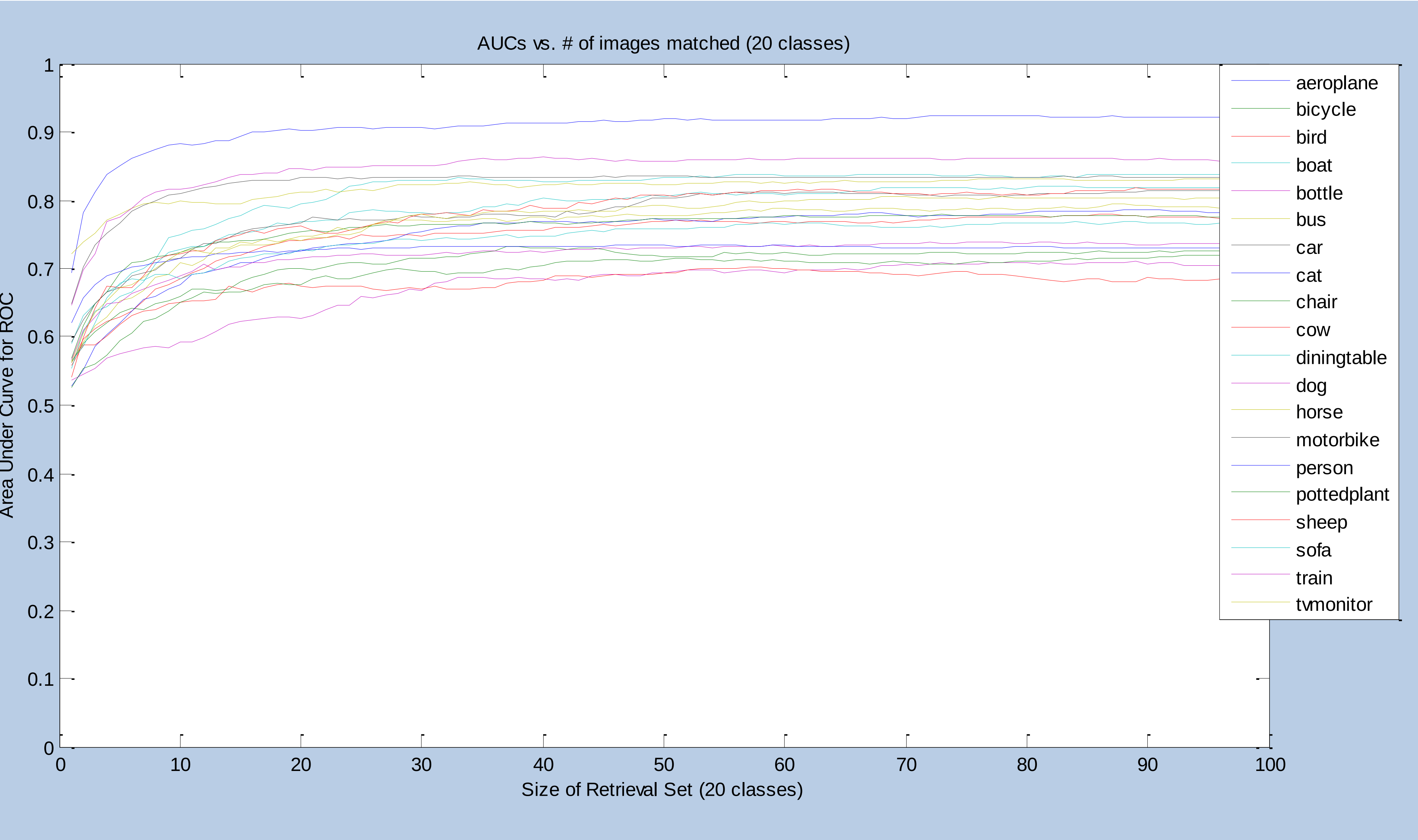
Pascal 2007 Results



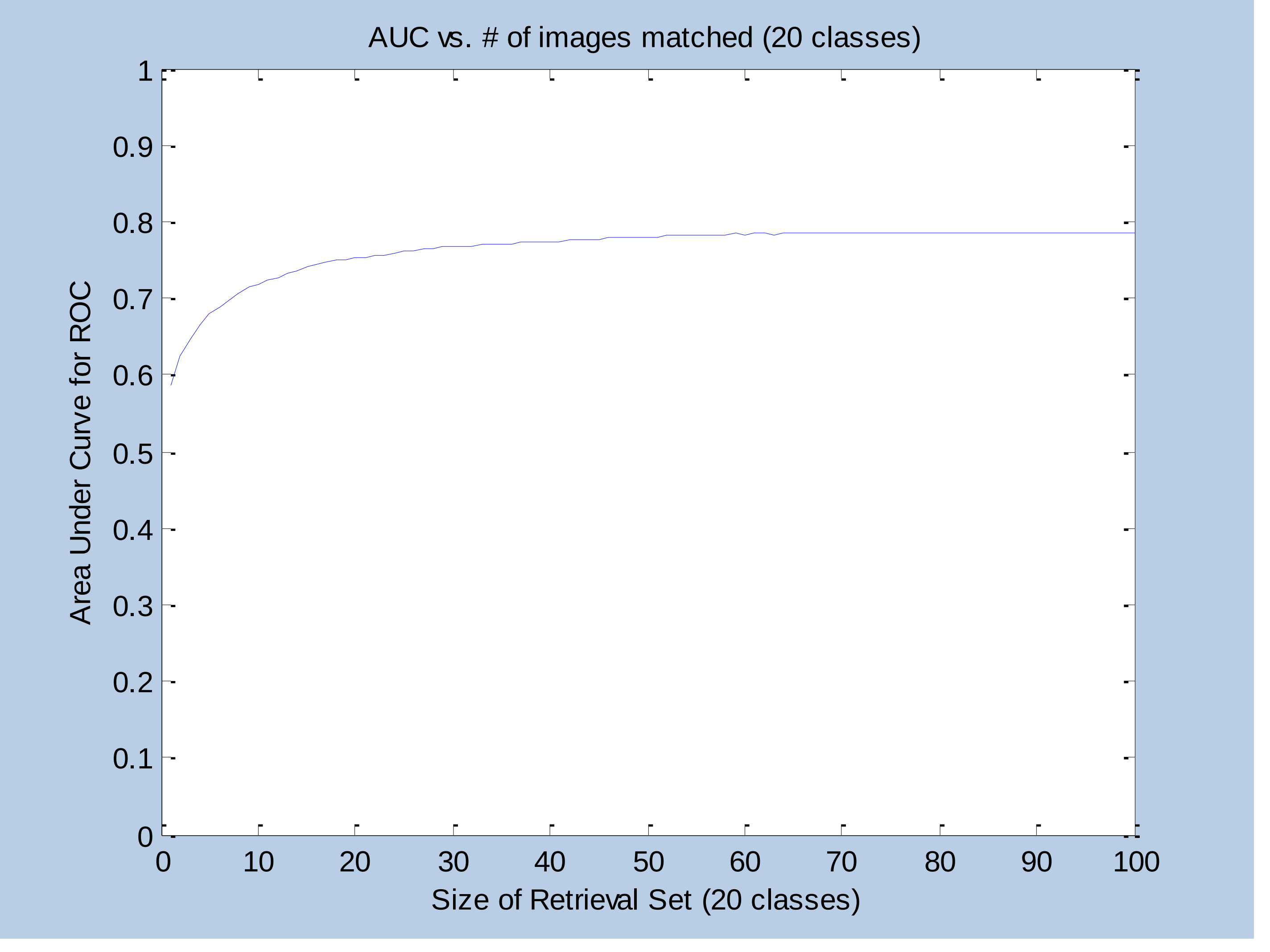
Pascal 2007 Results



Pascal 2007 Results



Pascal 2007 Results



Summary

- Stealing is good and helps your accuracy

- SIFT Flow tries to solve the finite data problem
  - Morph images so they do match perfectly
  - Decent idea, but needs more work

- Context transfers info from similar images
  - Small but noticeable improvements
  - How much data do you need?

Conclusion

- Context is yet another knob to tweak