Discriminative Clustering for Image Co-Segmentation

Joulin, A.; Bach, F.; Ponce, J. (CVPR. 2010)

Iretiayo Akinola
Josh Tennefoss

Outline

- Why Co-segmentation?
- Previous Work
- Problem Formulation
- Experimental Results
- Discussion: How do we improve?

Why Co-segmentation?

Segmentation (**Regions** & **Boundaries**)

- Foreground- background (2-regions)

Unsupervised Segmentation is Hard!

- Interactive segmentation
- Co-segmentation

Why Co-segmentation?

Unsupervised Segmentation is Hard!

- **Interactive segmentation** (Bounding box, Intelligent scissors, region seeds)

Why Co-segmentation?

Unsupervised Segmentation is Hard!

- Interactive segmentation
- **Co-segmentation** (with object names tagged)

Previous Work: Rother et al. 2006

"Cosegmentation of Image Pairs by Histogram Matching - Incorporating a Global Constraint into MRFs"

A generative model for cosegmentation that minimizes energy function:

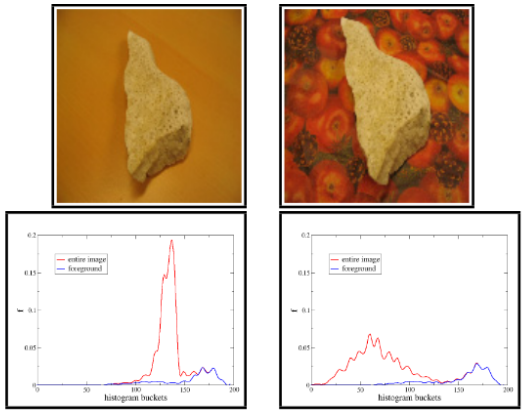
$$E(\mathbf{x}) = E_1(\mathbf{x}) + E_2(\mathbf{x})$$

$E_1(x)$: encodes spatial coherency

$E_2(x)$: penalizes differences from fg/bg models

Co-segmentation now formulated as optimization problem

- graph-cut technique applied



Previous Work: Hochbaum & Singh, 2009

An efficient algorithm for Co-segmentation

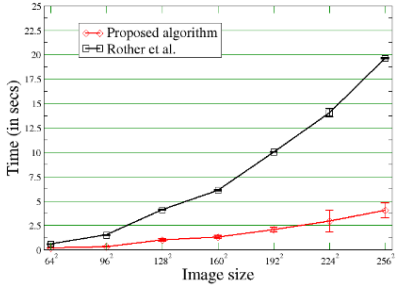
Combines object recognition and image segmentation; pick out objects and their segmentations (detects multiple segments)

Similar set-up with an energy function: $E(\mathbf{x}) = E_1(\mathbf{x}) + E_2(\mathbf{x})$

$E_1(x)$: an MRF term encoding spatial coherency and

$E_2(x)$: **maximizes the similarity between similar regions**.

polynomial time optimization algorithm

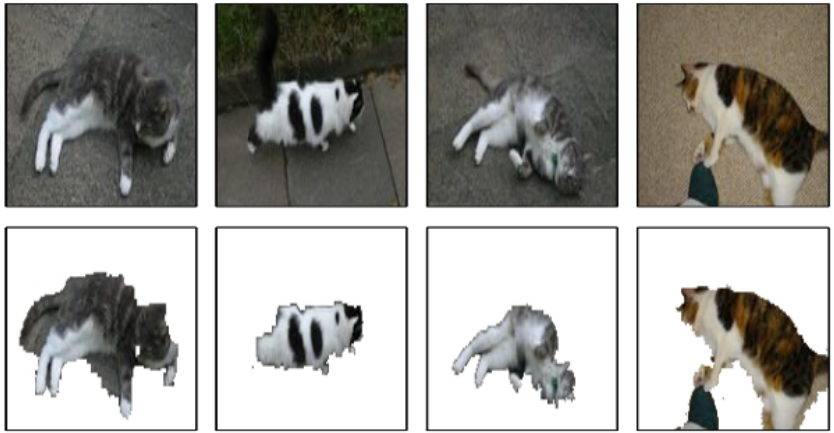


Common Ideas

- Image data share common material
- Similar pixels in different images should be assigned to the same class
- Different regions should have different generative models
- Energy function that combines these two ideas
- Optimization techniques for minimizing energy function

Contribution of this paper

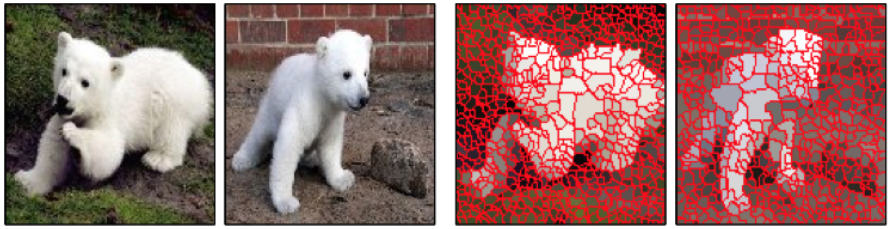
More robust co-segmentation algorithm that performs well on a larger range of foreground appearances.



Overview and Goals

- **Goal**: Co-segment images better than segmenting a single image alone
- **Output**: Y is in {-1, 1} for all pixels, corresponding to foreground or background
- **Idea**: If a pixel is foreground in one image, and there is a similar pixel (spectrally, spatially, and in feature space) in another image, then it is more likely to be foreground in that image. Similarly for background.
- **Benefit**: We have lots of tagged images (e.g. ImageNet)

Overview of Approach



First step is over-segmentation (derive super-pixels) so the algorithm can fit within reasonable memory and compute requirements.

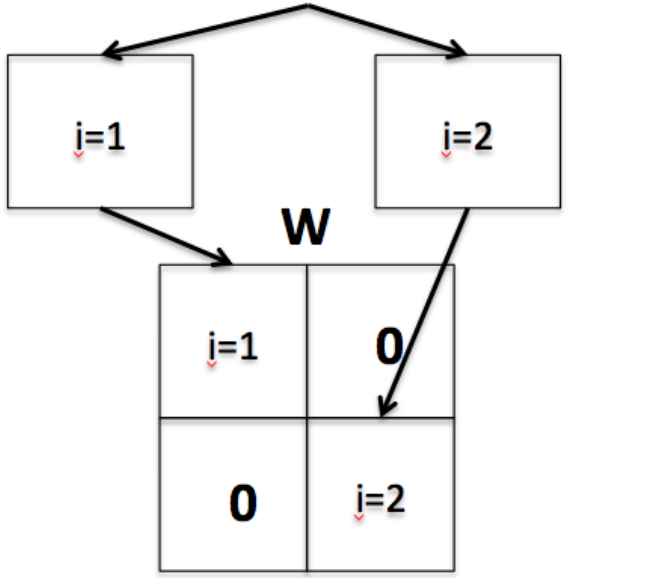
To go from over-segmentation to the real-valued mask:

1. Account for spatial relationships within a single image (maximize appearance consistency)
2. Extract features for each pixel (SIFT, gabor, color histogram)
3. Adjust for the two images having similar foreground objects (discriminate the common foreground pixels from other regions)

Formulation – Spatial Agreement

- Create a block diagonal similarity matrix, W, based on colors, c, and positions, p, on close pixels

$$W_{lm}^{i} = \exp(-\lambda_p \|p^m - p^l\|^2 - \lambda_c \|c^m - c^l\|^2)$$



Formulation – Spatial Agreement

- Define the normalized Laplacian as

$$L = I_n - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

- If we output second smallest eigenvector, we would

$$\begin{aligned} &\min y^T L y \\ s.t.\ \ &||y||^2 = n \\ &y^T D^{-\frac{1}{2}} 1_n = 0 \end{aligned}$$

- So include $y^T L y$ in objective

Formulation - Discriminative Agreement

- Create an SPD similarity matrix on all pixels using their feature space and their (currently unknown) labels.
- k features extracted
- l & m are the pixels, x_d is the feature

$$K_{lm} = \exp\left(-\lambda_h \sum_{d=1}^{k} \frac{(x_d^l - x_d^m)^2}{x_d^l + x_d^m} \right)$$

Formulation - Discriminative Agreement

- Phi is chosen based on the book *Kernel Methods for Pattern Analysis,* Share-Taylor and Christiann, 2004
- Fit a ML model to K, to best fit an affine classifier, f and b, to fit y based based on K

$$K_{lm} = \exp\left(- \lambda_h \sum_{d=1}^{k} \frac{(x_d^l - x_d^m)^2}{x_d^l + x_d^m} \right)$$

$$K_{ml} = \Phi(x^m)^T \Phi(x^l)$$ For some feature map Φ

$$\min_{f,b} \quad \frac{1}{n} \sum_{j=1}^{n} \ell(y_j, f^T \Phi(x^j) + b) + \lambda_k \| f \|^2$$

Formulation - Discriminative Agreement

Using DIFFRAC (Bach & Harchaoui 2007), for each (prediction) y, we can calculate **overall class inseparability as g(y)**:

$$g(y) = y^\top A y, \tag{4}$$

where $A = \lambda_k (I_n - \frac{1}{n} 1_n 1_n^T)(n\lambda_k I_n + K)^{-1}(I_n - \frac{1}{n} 1_n 1_n^T)$.

Formulation - Combining Spatial and Discriminative

Discriminative agreement between all pixels

$$\min_{y \in \{-1,1\}} y^T A y$$

Spatial agreement within a single image

$$\min_{y \in \{-1,1\}} y^T L y$$

$$\min_{y \in \{-1,1\}} y^T (A + \frac{\mu}{n} L) y$$

Restriction on Solution

- The trivial solution (where all pixels are of the same class) is optimal for A without an additional constraint.
- Add a constraint, for each image i, to force the percent of pixels from each class in each image to be bounded by λ_0 and λ_1 they use 5% and 95%, respectively.

$$\min_{y \in \{-1,1\}^n} y^T \left(A + \frac{\mu}{n} L\right) y, \tag{5}$$

$$\text{subject to} \quad \forall i,\ \lambda_0 n_i \delta_i \leqslant \tfrac{1}{2}(yy^\top + 1_n 1_n^T)\delta_i \leqslant \lambda_1 n_i \delta_i.$$

Solving: convex objective

$$\min_{y \in \{-1,1\}^n} y^T \left(A + \tfrac{\mu}{n} L\right) y, \qquad (5)$$

$$\text{subject to} \quad \forall i,\ \lambda_0 n_i \delta_i \leqslant \tfrac{1}{2}(yy^\top + 1_n 1_n^T)\delta_i \leqslant \lambda_1 n_i \delta_i.$$

Mixed-integer problem is not convex

For convexity, change from restricting y to be an integer to being in an elliptope

$$\mathcal{E} = \{Y \in \mathbb{R}^{n \times n}\ ,\ Y = Y^T\ ,\ \mathrm{diag}(Y) = 1_n\ ,\ Y \succeq 0\},$$

$$\min_{Y \in \mathcal{E}} \mathrm{tr}\left(Y\left(A + \tfrac{\mu}{n} L\right)\right), \qquad (6)$$

$$\text{subject to} \quad \forall i,\ \lambda_0 n_i \delta_i \leqslant \tfrac{1}{2}(Y + 1_n 1_n^T)\delta_i \leqslant \lambda_1 n_i \delta_i$$

$$\mathrm{rank}(Y) = 1.$$

But rank(Y) = 1 is not convex ☹

Solving: low rank solution

- Journee Et Al (2008) have published a lower rank semidefinite problem solver. But, this solver cannot handle inequality constraints….
- So for each constraint, h, add a twice differentiable penalty of

$$\frac{\nu}{\alpha}\log(1+\exp(\alpha h(Y))$$

- With some (excluded) tricks, we can find relatively low rank solutions over Y.

Getting y

- Project the solution Y onto unit rank PD matrices by taking the eigenvector of the largest eigenvalue, giving $y \in \mathbb{R}^n$

Getting y

- Then round each element of y to 1 or -1 relative to 0.
- Hand select if 1 is foreground or background, and -1 is the other one

Experimental Results

- Low-variability datasets

  very similar foreground, fewer images

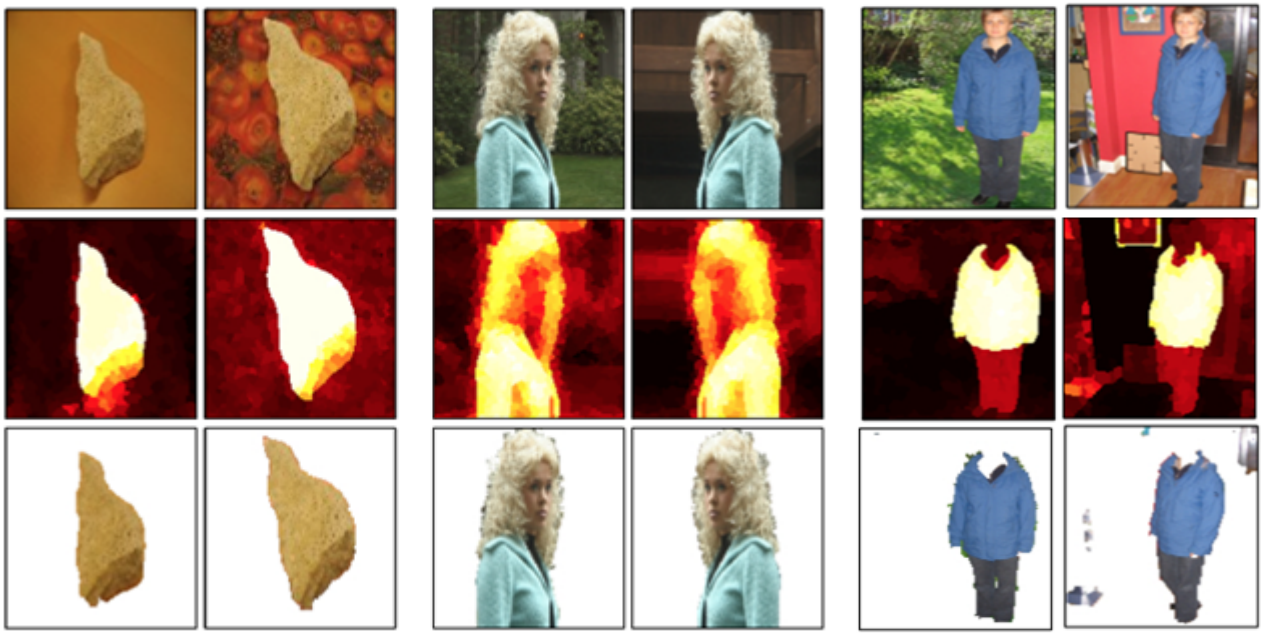
- High-variability datasets

  higher variation in foreground appearance, more images to co-segment

metric: misclassification error (% of well-classified pixels in each image.)

Experimental Results

Low Inter-class variation



Results Summary

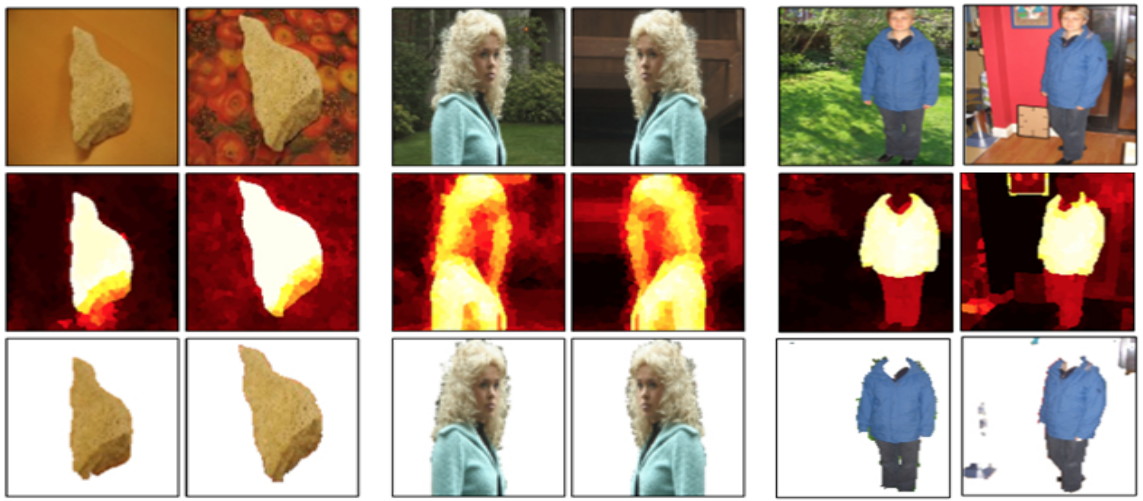
Low Inter-class variation

	Girl	Stone	Boy	Bear	Dog
our method	**0.8** %	**0.9** %	6.5 %	5.5%	6.4 %
[12]	-	1.2%	**1.8** %	**3.9** %	**3.5** %

Table 1. Segmentation accuracies on pairs of images.

Experimental Results

Low Inter-class variation



	Girl	Stone	Boy	Bear	Dog
our method	**0.8** %	**0.9** %	6.5 %	5.5%	6.4 %
[12]	-	1.2%	**1.8** %	**3.9** %	**3.5** %

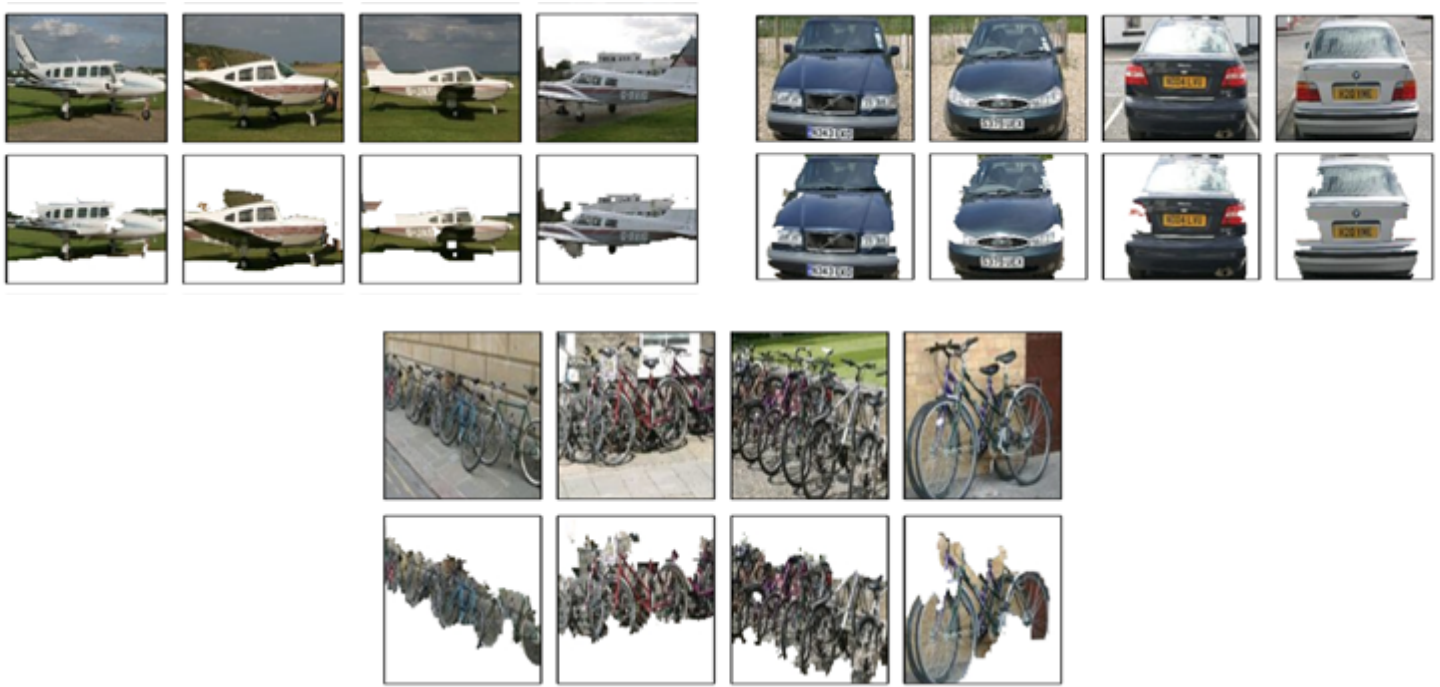
Table 1. Segmentation accuracies on pairs of images.

Experimental Results

High Inter-class variation

Experimental Results

High Inter-class variation



Results Summary

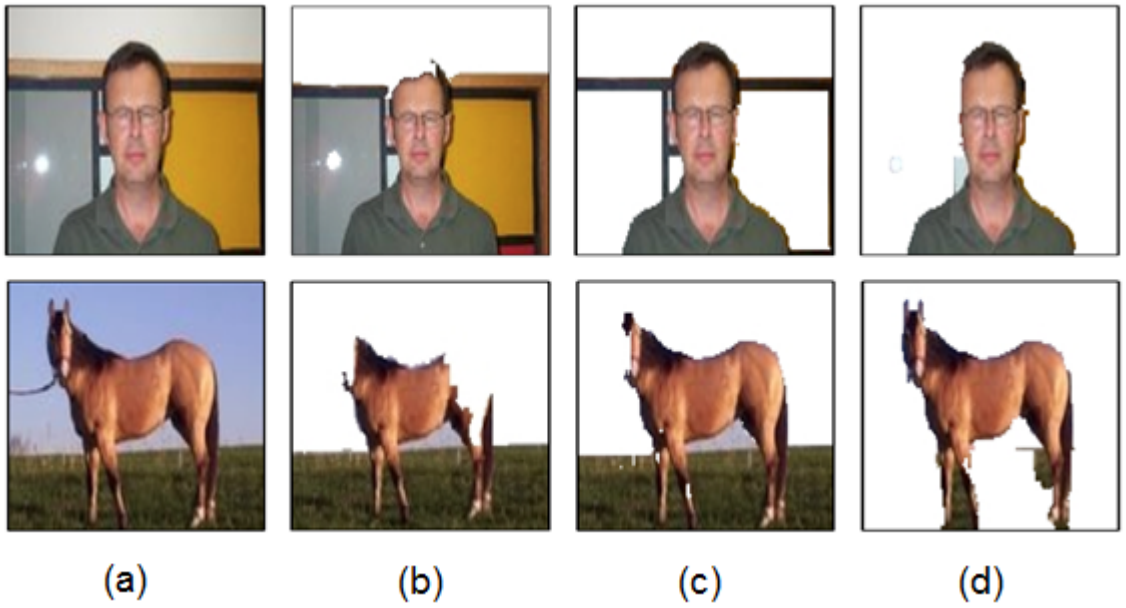
High Inter-class variation

class	images	our method	single-image	MNcut [7]	uniform	μ
Cars (front)	6	87.65% ±0.1	**89.6 % ±0.1**	51.4 % ±1.8	64.0 % ±0.1	1
Cars (back)	6	**85.1 % ±0.2**	83.7 % ±0.5	54.1%±0.8	71.3 % ±0.2	1
Face	30	**84.3% ±0.7**	72.4% ±1.3	67.7% ±1.2	60.4% ±0.7	1
Cow	30	**81.6 % ±1.4**	78.5 % ±1.8	60.1% ±2.6	66.3 % ±1.7	0.001
Horse	30	**80.1 % ±0.7**	77.5 % ±1.9	50.1% ±0.9	68.6 % ±1.9	0.001
Cat	24	**74.4 % ±2.8**	71.3 % ±1.3	59.8% ±2.0	59.2 % ±2.0	0.001
Plane	30	73.8 % ±0.9	62.5 % ±1.9	51.9% ±0.5	**75.9 % ±2.0**	0.001
Bike	30	**63.3 % ±0.5**	61.1 % ±0.4	60.7% ±2.6	59.0% ±0.6	0.001

Table 2. Segmentation accuracies on the Weizman horses and MSRC databases.

Why Co-segmentation?

multi-image vs single-image segmentation



(a) Original Image
(b) multiscale normalized cut
(c) our algorithm on a single image
(d) our algorithm on 30 images

Discussion:
How might we improve this?

- **Accuracy**

Holistic image similarity metric before running this method, i.e. weight the K matrix according to the image from which the pixel is taken

- **Accuracy**

Different features for similarities, e.g. autoencoder or CNN-based features

Discussion:
How might we improve this?

- **Border resolution**

Rerun seeded algorithm without superpixels on the fg/bg boundary

- **Leverage other technologies**

Output a bounding box or seeds → GrabCut or similar

- **What do you think???**