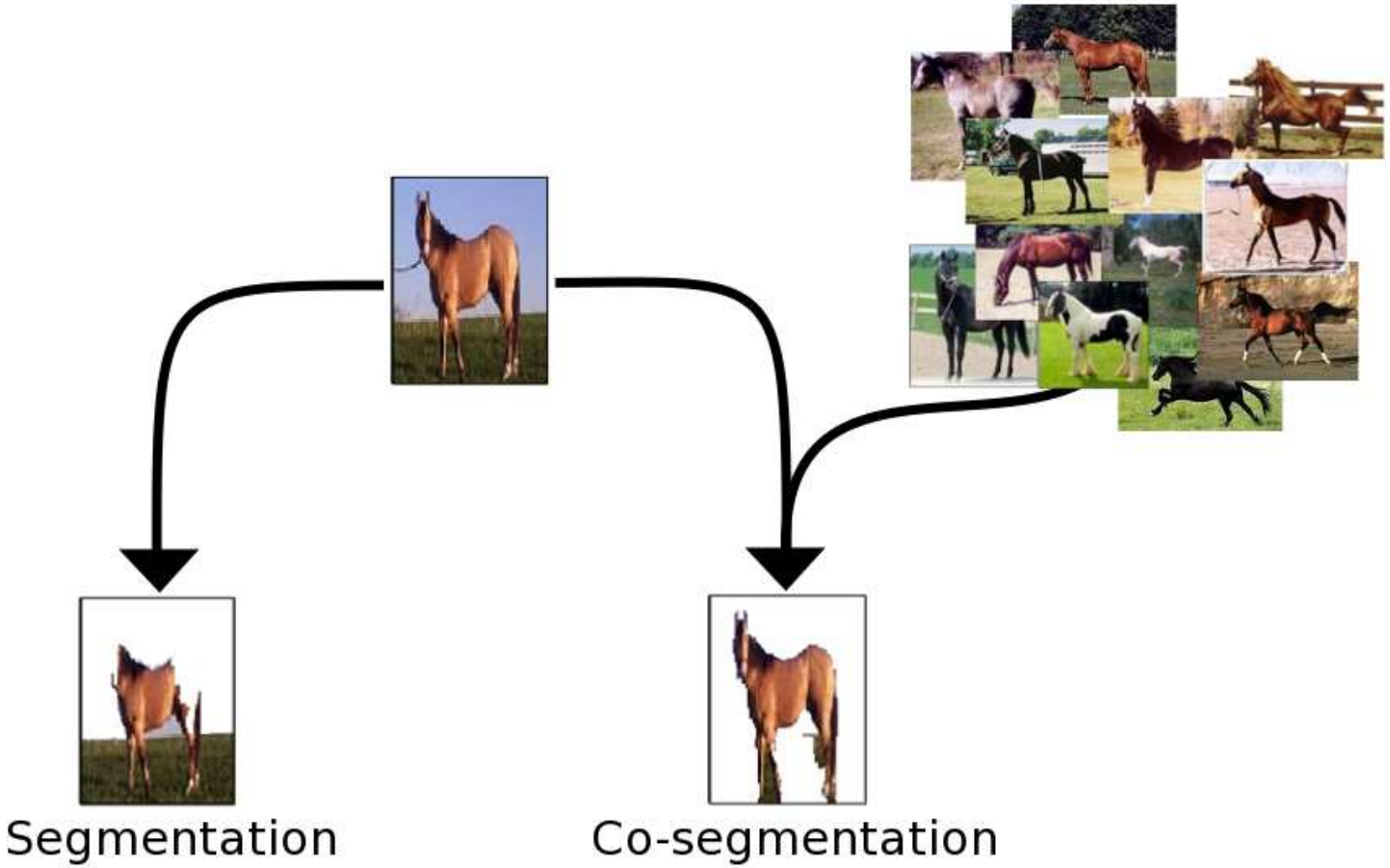
Discriminative Clustering for Image Co-segmentation

Armand Joulin[1,2] Francis Bach[1,2] Jean Ponce[2]

[1]INRIA [2]Ecole Normale Supérieure

January 15, 2010

Introduction



Introduction

- problem: dividing simultaneously q images in k different regions (segments)
- when $k = 2$, this reduces to dividing images into foreground and background regions.
- Our approach solves simultaneously the object recognition problem and the segmentation

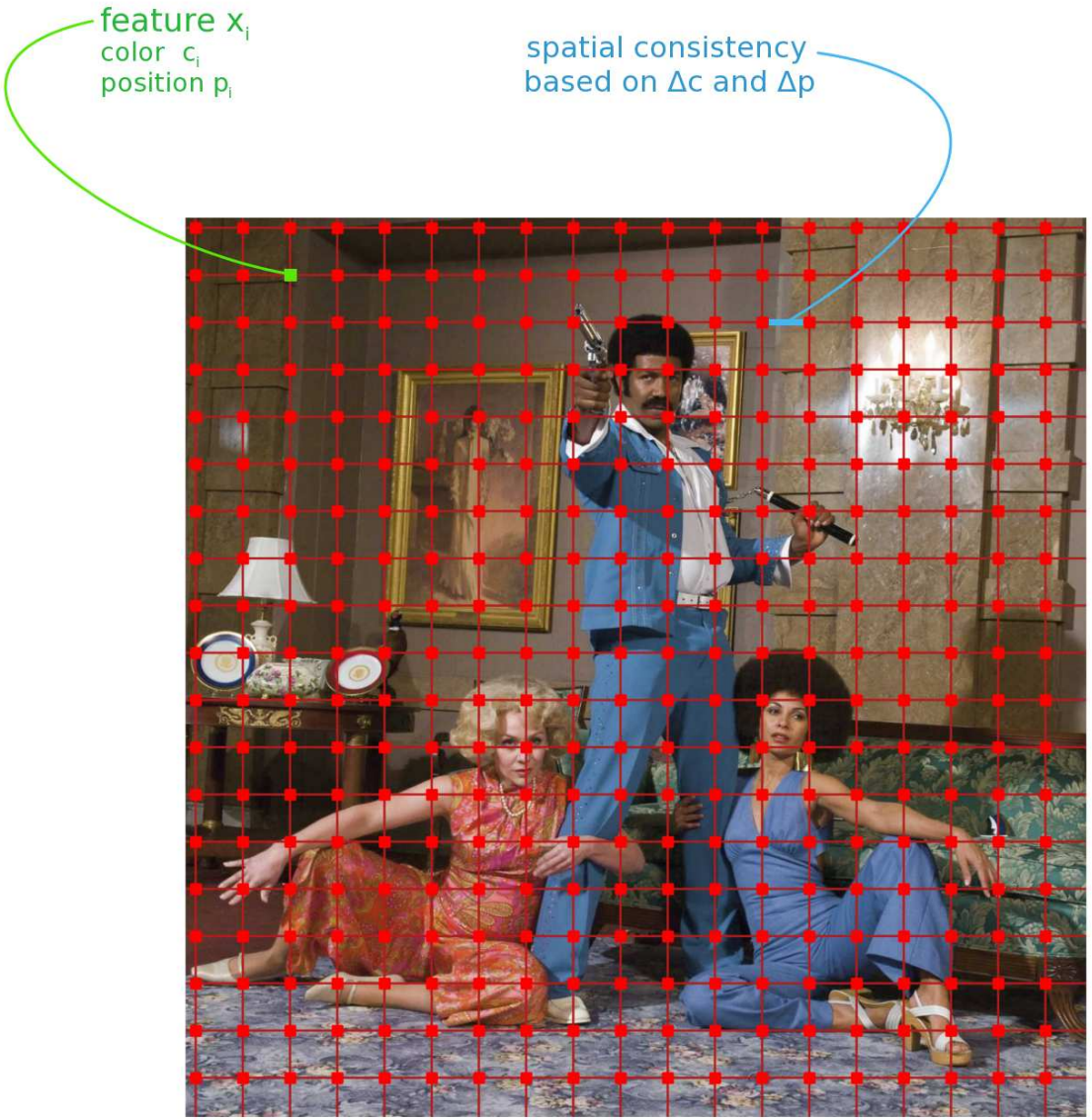
Introduction

- Our framework: unsupervised discriminative clustering.
- It is well-adapted to segmentation problems for 2 reasons :
  - it allows to re-use existing features for supervised classification or detection
  - it allows to easily introduce spatial and local color-consistency constraints.

Problem Notations

- q images. Each image i is reduced to a subsampled grid of n_i pixels
- For the j-th pixel, we denote by :
  - $c^j \in \mathbb{R}^3$ its color,
  - $p^j \in \mathbb{R}^2$ its position within the corresponding image,
  - x^j an additional k-dimensional feature vector.
- As $k = 2$, we note y the vector such as $y_j = 1$ if the i-th pixel is in the foreground and -1 otherwise.

Problem Notations



Local consistency and discriminative clustering

- co-segmenting images to find a common object relies on two tasks :
  - maximizing the separability of two classes between different images
  - maximizing spatial and appearance consistency within a particular image

Local consistency

- Spatial consistency *within* an image i is enforced through a similarity matrix W^i
  - W^i is based on color features (c^j) and spatial position (p^j) through a Gaussian model
  - The similarity between two pixels l and m within an image i is thus:

$$W^i_{lm} = \exp(-\lambda_p \|p^m - p^l\|^2 - \lambda_c \|c^m - c^l\|^2), \qquad (1)$$

- Finally on consider the Laplacian matrix L based on the block-diagonal matrix W (with W_i on its diagonal) :

$$L = I_n - D^{-1/2} W D^{-1/2} \qquad (2)$$

Discriminative clustering

- Our discriminative clustering framework is based on positive definite kernels
- Our features are histograms therefore we use a kernel matrix K based on the χ^2-distance:

$$K_{lm} = \exp\left(-\lambda_h \sum_{d=1}^{k} \frac{(x_d^l - x_d^m)^2}{x_d^l + x_d^m} \right), \tag{3}$$

- It is equivalent to mapping each of our n k-dimensional vectors x^j, $j = 1, \ldots, n$ into a high-dimensional Hilbert space $\mathcal{F}$ through a feature map Φ, so that $K_{ml} = \Phi(x^m)^T \Phi(x^l)$

Discriminative clustering

- We aim to fit a linear model of Φ to the labels y :

$$g(y) = \frac{1}{n}\sum_{j=1}^{n} \ell(y_j, f^T\Phi(x^j) + b) + \lambda_k \|f\|^2, \tag{4}$$

  where ℓ is a loss function.
- Taking a square loss function ($\ell(a,b) = (a-b)^2$), the solution of the problem above leads to a close-form formulation:

$$g(y) = \mathrm{tr}(Ayy^T)$$

  where $A = \lambda_k(I_n - \frac{1}{n}1_n1_n^T)(n\lambda_k I_n + K)^{-1}(I_n - \frac{1}{n}1_n1_n^T)$.

cluster size constraints

- Putting all pixels into a single class leads to perfect separation
- Therefore, we add constraints on the number of elements in each class
- we constrain the number of elements of each class in each image to be upper bounded by λ_1 and lower bounded by λ_0.
- we note $\delta_i \in \mathbb{R}^n$ is the indicator vector of the i-th image

Problem formulation

- Combining the spatial consistency through Laplacian matrix L,the discriminative cost through the matrix A and the cluster size constraints, we obtain the following problem :

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

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

Optimization - Convex Relaxation

- We can reparameterize our problem with $Y = yy^T$. Y is usually referred to as the equivalence matrix
- Y is symmetric, positive semidefinite, with diagonal equal to one, and unit rank.

Optimization - Convex Relaxation

- if we denote by $\mathcal{E}$ the *elliptope*, i.e., the convex set defined by:

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

  Our optimization problem becomes :

$$\begin{aligned} & \min_{Y\in\mathcal{E}} \operatorname{tr}\big(Y\big(A + \frac{\mu}{n}L\big)\big), \\ \text{subject to} \quad & \forall i,\ \lambda_0 1_n \leqslant (Y + 1_n 1_n^T)\delta_i \leqslant \lambda_1 1_n \\ & \operatorname{rank}(Y) = 1. \end{aligned}$$

- Because of the rank constraint, This is not a convex optimization problem. The convex relaxation of our problem is to remove the rank constraint.

Optimization

- We optimize a convex problem over positive definite matrices, usually referred to as a semidefinite program (SDP)
- Without using the structure of this problem, general purpose toolboxes would solve this problem in $O(n^7)$
- Bach and Harchaoui [**?**] (2007) consider a partial dualization techniquethat scale up to thousands of data points.
- To gain another order of magnitude, we consider the optimization through low-rank matrices considered by Journée et al (2008).

Efficient low-rank optimization

- We use the procedure developed in Journée et al (2008).
- However this procedure cannot deal with our cluster size inequalities.
- Therefore we use a augmented Lagrangian method to transform these constraints into penalties
- for each inequality constraint $h(Y)$, we add a a twice differentiable convex penalty term to the objective function, in our case $\nu \max\{0, h(Y)^3\}$.

Efficient low-rank optimization

- We are now faced with the optimization of a convex function $f(Y)$ on the elliptope $\mathcal{E}$.
- It leads to low-rank solutions
- If the rank of the solution r is not known, Journée et al. (2008) have designed an adaptive procedure.
- We optimize the function $g_d : y \mapsto f(yy^\top)$ defined for $y \in \mathbb{R}^{n \times d}$ such that $\mathrm{diag}(yy^\top) = 1$.

Journée et al. low-rank optimization (2008)

- first notice that the cost g_d is invariant by right-multiplication of y by a $d \times d$ orthogonal matrix.
- Therefore we perform our minimization on the quotient space $\bar{\mathcal{E}}_d = \mathcal{E}_d / \mathcal{O}_d$, where $\mathcal{E}_d = \{Y \in \mathcal{E}, \mathrm{rank}(Y) = d\}$, where $\mathcal{E}_d = \{Y \in \mathcal{E}, \mathrm{rank}(Y) = d\}$ and $\mathcal{O}_d = \{P \in \mathbb{R}^{d \times d} | PP^T = I_d\}$.
- It is a Riemannian manifold.
- In order to find a local minimum on this quotient space, we can thus use a trust-region method for such manifolds

Preclustering

- Since our cost function f uses a full $n \times n$ matrix $A + (\mu/n)L$, our algorithm may require too much memory for storing it.
- To reduce the total number of pixels, we use superpixels obtained from an oversegmentation of our images.

Rounding

- In order to retrieve $y \in \{-1, 1\}$, from our relaxed solution, we compute the largest eigenvector $e \in \mathbb{R}^n$ of Y.
- Then our final clustering is $y = \text{sign}(e)$.

method overview

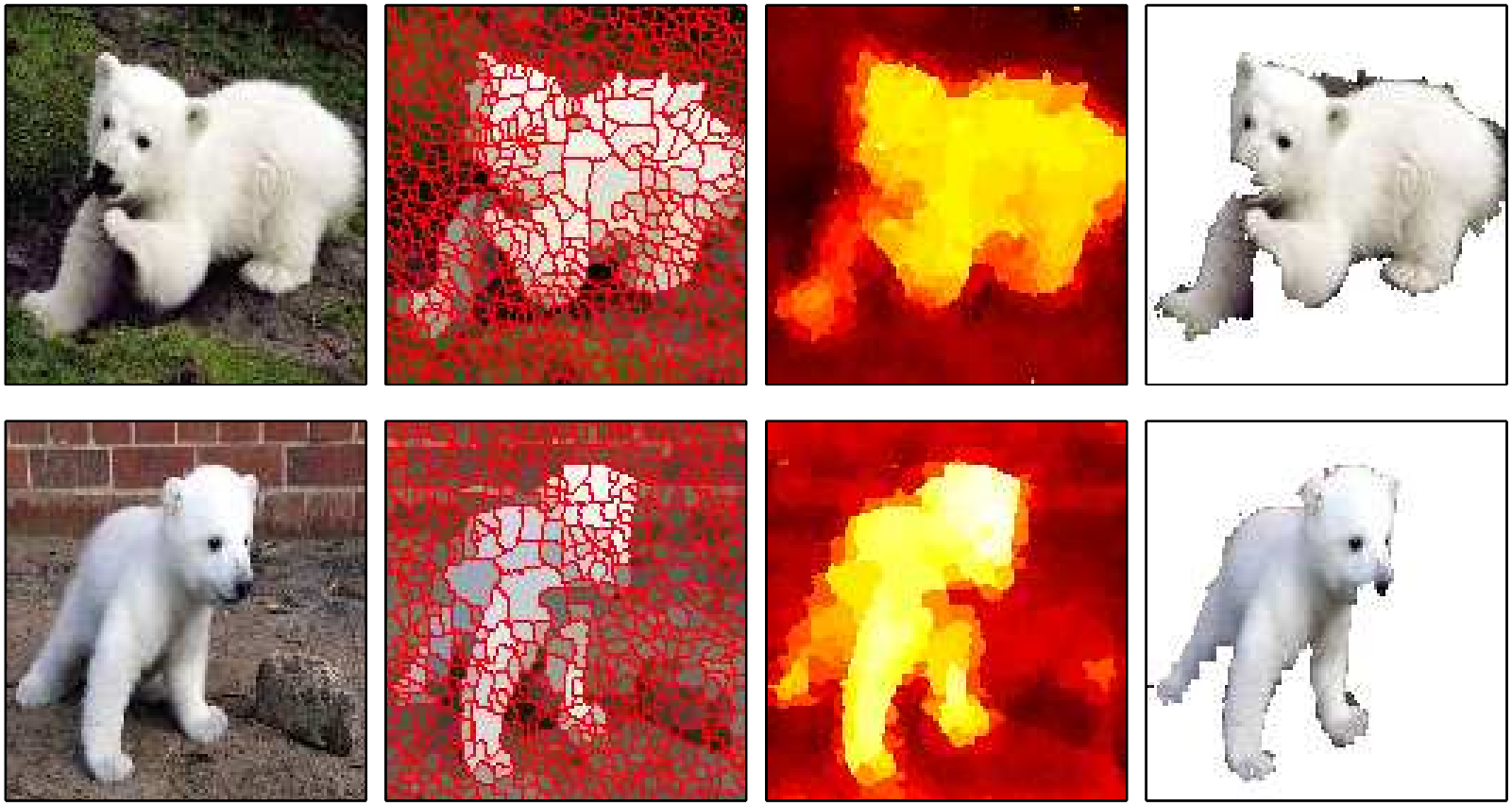


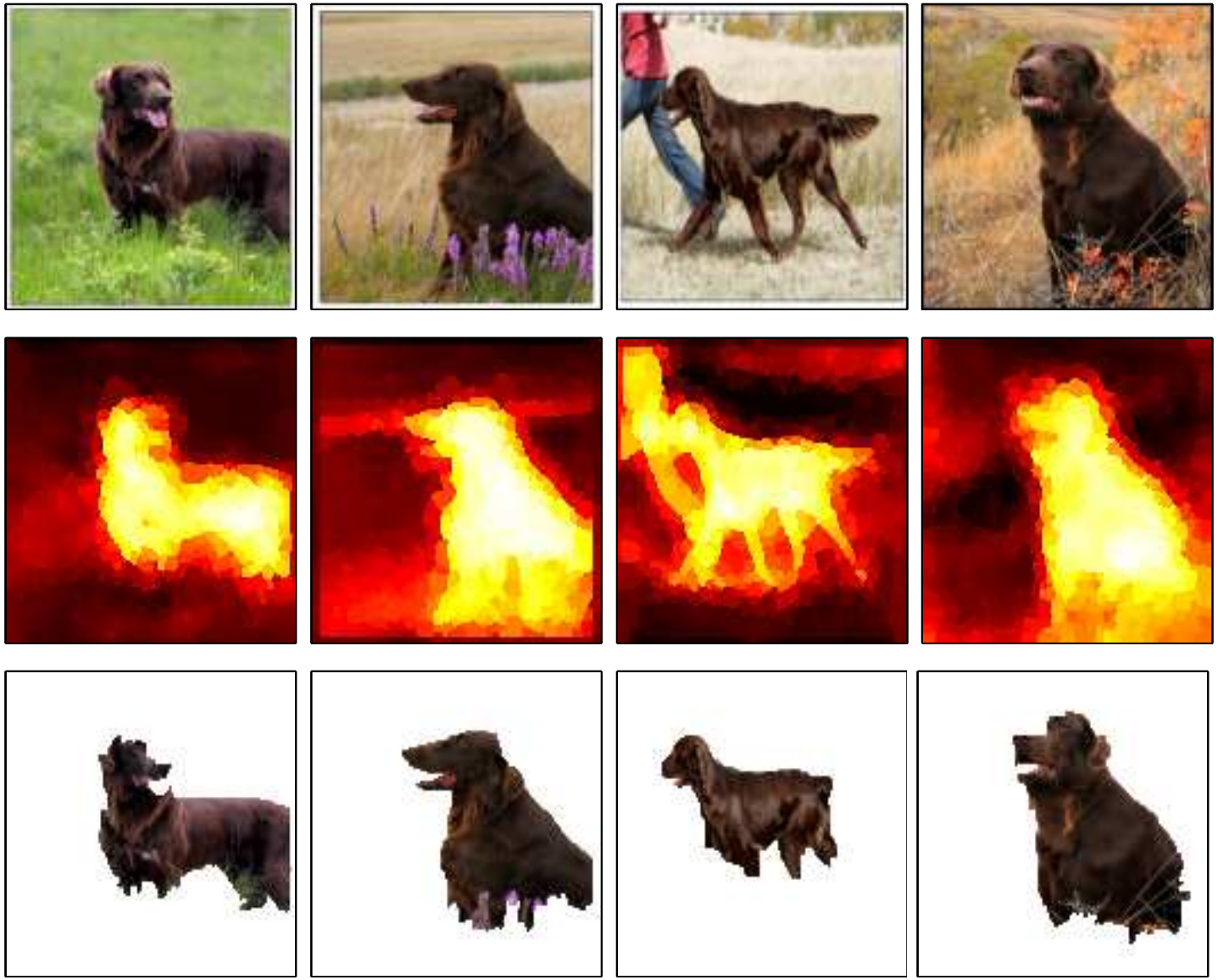
Figure: Illustrating the co-segmentation process on two bear images; from left to right: input images, over-segmentations, scores obtained by our algorithm and co-segmentations. $\mu = 0.1$.

Results

results on two different problems :

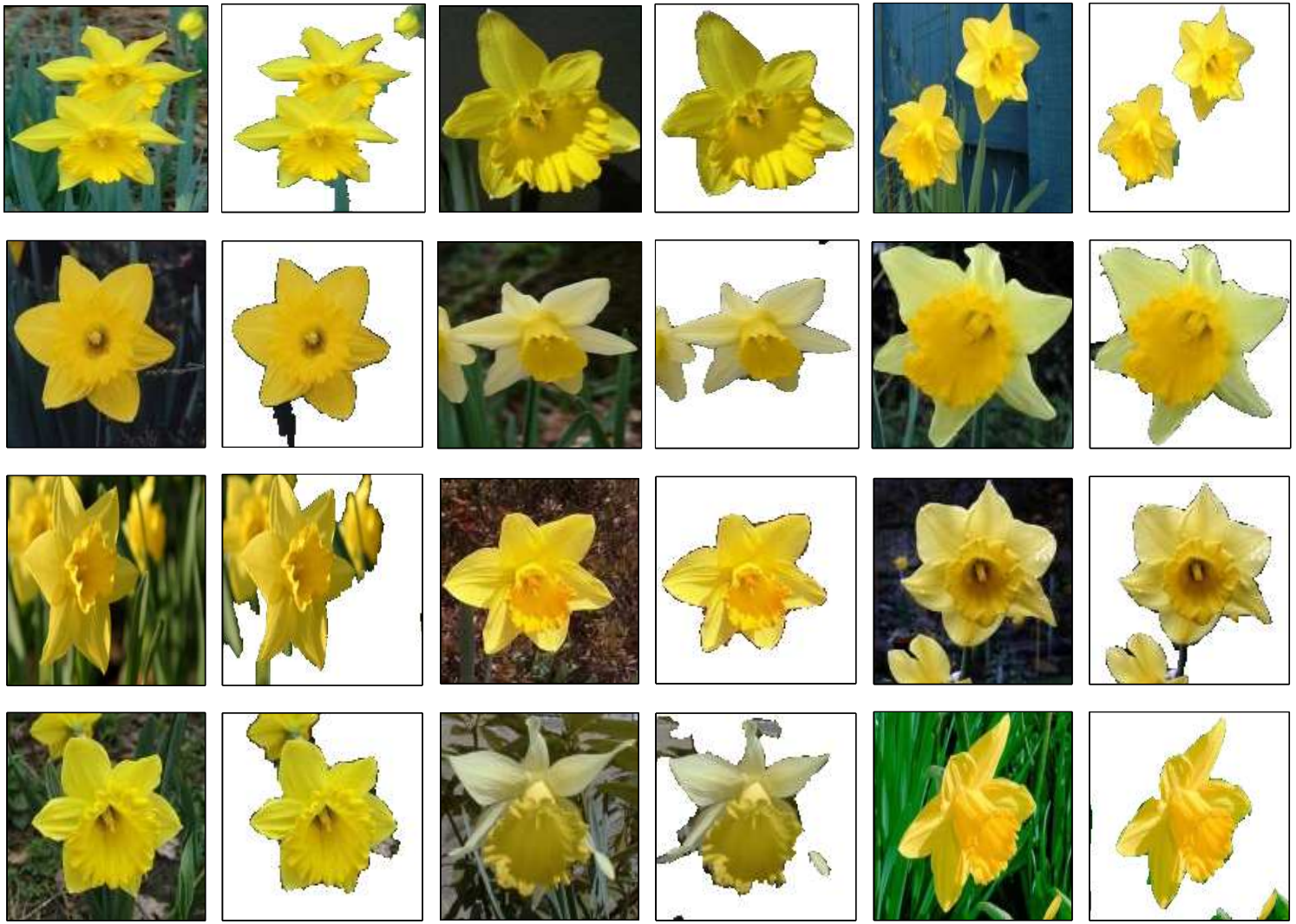
- we first consider images with foreground objects which are identical or very similar in appearance and with few images to co-segment,
- Then, we consider images whose foreground objects exhibit higher appearance variations and with more images to co-segment (up to 30).

Results - similar object



Results - similar object

Results - similar object



Results - similar class of object

Figure: Faces. On the bottom right, an example on which the algorithm fails.

Results - similar class of object

Figure: Cows. On the bottom right, an example on which the algorithm fails.

Results - similar class of object

Figure: Horses. On the bottom right, an example on which the algorithm fails.

Results - similar class of object

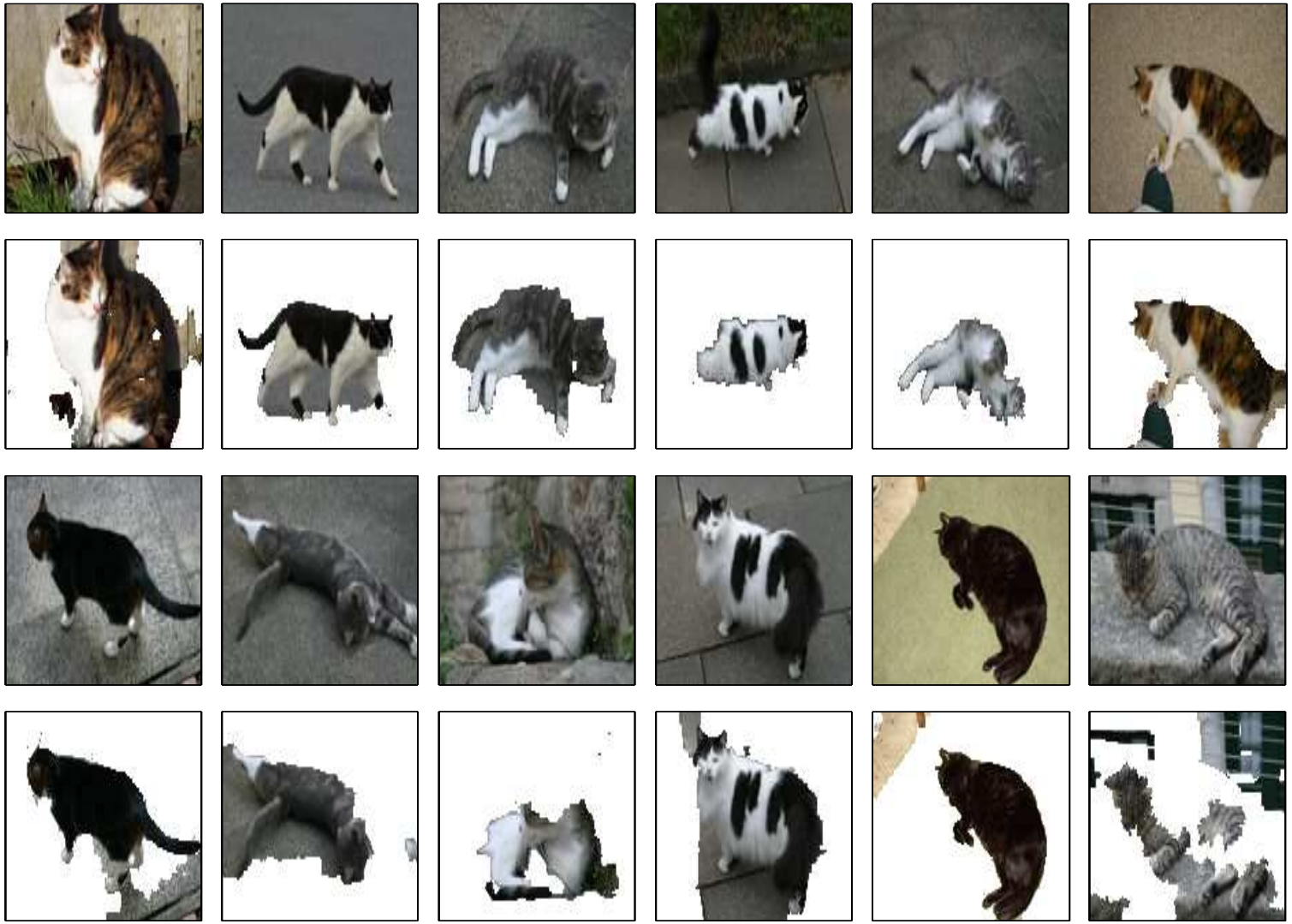


Figure: Cats. On the bottom right, an example on which the algorithm fails.

Results - similar class of object

Figure: Bikes. On the bottom right, an example on which the algorithm fails.

Results - similar class of object

Figure: Planes. On the bottom right, an example on which the algorithm fails.

Comparison with N-cut

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

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

Comparison with N-cut

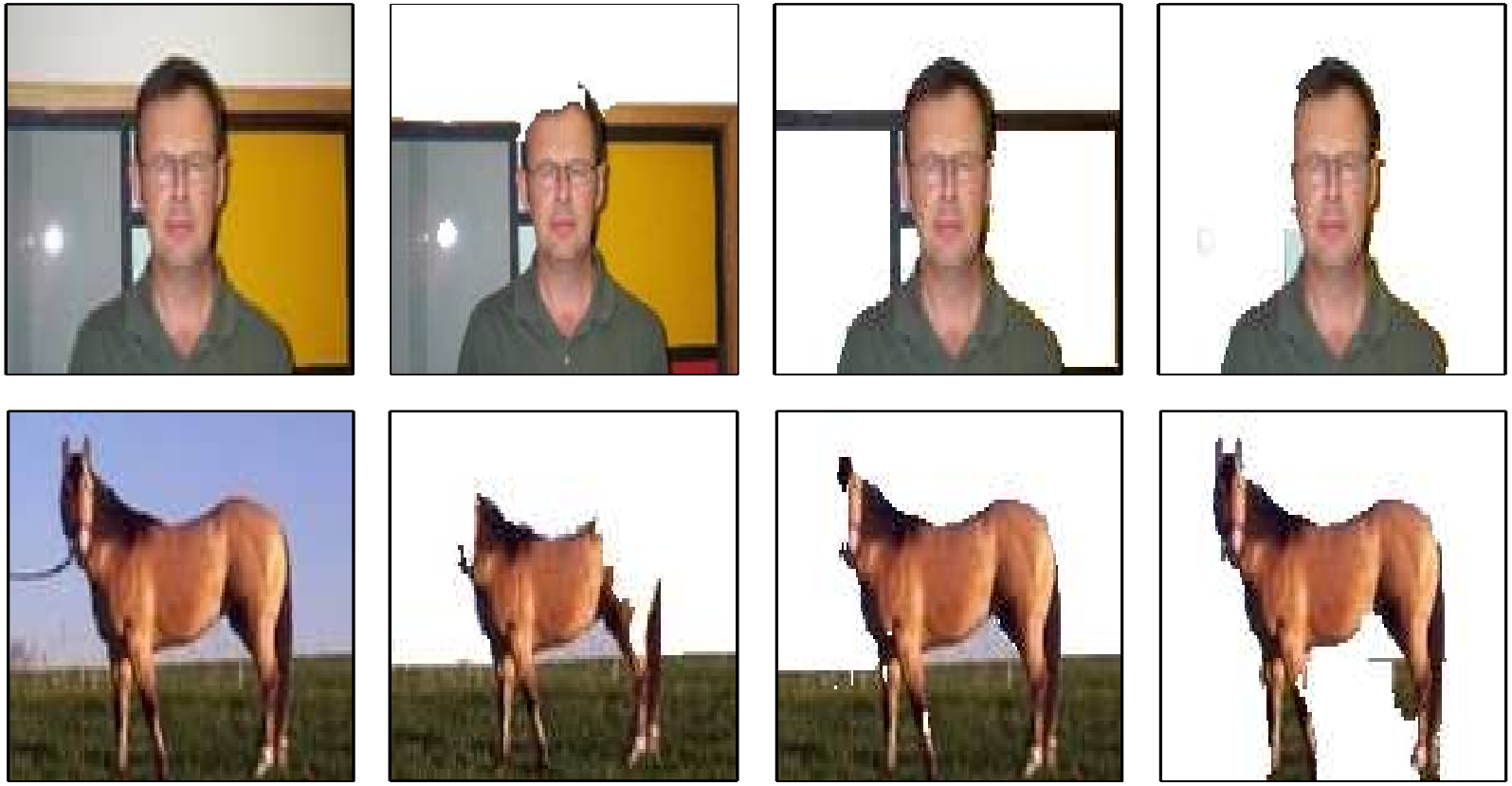


Figure: Comparing co-segmentation with independent segmentations; from left to right: original image, multiscale normalized cut, our algorithm on a single image, our algorithm on 30 images.

Perspective

- Consider more than 2 classes
- feature selection
- scale up to hundred of thousands
- change the loss function